FACULTAD DE INGENIERÍA Y COMPUTACIÓN

Escuela Profesional de Ciencia de la Computación

"Improving the ILS-TQ technique for The High School Timetabling Problem"

Presentado por:

Jonathan Roger Alvarez Ponce

Para Optar por el Título Profesional de:
Licenciado en Ciencia de la Computación

Asesor: Dr. Alexander Javier Benavides Rojas

Arequipa, Agosto de 2019

Dedicado a mi familia que me ha cuidado y apoyado siempre, a todos los profesores por haberme inculcado de los conocimientos necesarios, a mis amigos por haberme ayudado cuando lo necesitaba y a mi asesor por darme una buena orientación.

Acronyms

HSTTP *High School TimeTabling Problem*

SA Simulated Annealing

LNS Large Neighborhood Search

TS Tabu Search

GA Genetic Algorithm

VNS Variable Neighborhood Search

VND Variable Neighborhood Descent

ILS Iterated Local Search

HS Harmony Search

KHE Kingston High school timetabling Engine

CSO Cat Swarm Optimization

RVNS Reduced Variable Neighborhood Search

SVND Sequential Variable Neighborhood Descent

SVNS Skewed Variable Neighborhood Search

DGA Direct Genetic Algorithm

IGA Indirect Genetic Algorithm

ITC International Timetabling Competition

GVNS General Variable Neighborhood Search

PSO Particle Swarm Optimization

TQ Torque

MT Matching

ILS-TQ Iterated Local Search - Torque

2TQ Two TQ moves

RR Relaxed Rule

SC Random Swaps per Class

CC Change Random Column

S3R Swap Three Requirements

ARD Average Relative Deviation

SAI Simulated Annealing Inner

SAO Simulated Annealing Outer

Agradecimientos

En primer lugar, deseo agradecer a Dios por haberme guiado a lo largo de estos cinco años de estudio y en todo lo que he realizado y realizaré a lo largo de mi vida.

Asimismo, agradezco a mi familia que me ha apoyado en todo lo posible para forjarme como un profesional.

También quiero agradecer a mi asesor Prof. Dr. Alexander Javier Benavides Rojas por haberme orientado correctamente para la realización de esta tesis, brindándome lo necesario para no perderme en el camino.

Por último, agradezco a la universidad, mi alma matter, por haberme brindado la formación que me permitirá a mejorar la sociedad. Asimismo, agredecer a mis profesores que me han inculcado, tanto directa como indirectamente, de los conocimientos necesarios para ejercer mi profesión. También agradezco al personal administrativo de la misma universidad por velar que todo esté en orden.

Resumen

El problema de planificación de horarios para colegios de secundaria es un problema NP-Complete que consiste en asignar cursos, que son enseñados por profesores y asignados a cada clase, en periodos mientras se satisface restricciones. A través de los tiempos, las meta-heurísticas han dado mejores resultados para instancias reales de estos problemas que los métodos determinísticos ya que el espacio de búsqueda de los problemas de planificación son inmensos por lo que explorarlas todas resulta imposible. Mientras mejor sean los horarios, mayor es el rendimiento para los alumnos y profesores, además que reducen los costos para generar estos horarios. Se proponen modificaciones, por separado, a la búsqueda local iterada (ILS) con el operador Torque (TQ) para las 34 instancias reales de colegios de Brazil. Estas modificaciones por separado cambian cómo un horario es modificado y cómo este horario es aceptado. Nuestra implementation del esquema de enfriamiento del templado simulado, con algunas configuraciones de parámetros, ha dado mejores resultados que nuestros otros métodos y soluciones más consistentes que el método original para algunas instancias. Además, para crear otras instancias más fácilmente, se ha creado un formulario.

Palabras clave: Colegios de secundaria, Problema de planificación de horarios para colegios de secundaria, meta-heurísticas, Búsqueda Local Iterada, Operador Torque, creador de instancias.

Abstract

The High School Timetabling Problem is an NP-Complete problem that consists in allocating subjects, that are taught by teachers and assigned to each class, to periods while satisfying constraints. Throughout the years, meta-heuristics haven given better results to real-life instances compared to deterministic methods since the search space for timetabling problems are huge and exploring it completely is impossible. The better the schedules are, the better the students and teachers' performance, and the costs of generating these schedules are reduced. This proposal consists in modifications done separately to the Iterated Local Search (ILS) with the Torque (TQ) operator for the 34 real-life instances of schools of Brazil. These separate modifications change how a schedule is modified and how it is is accepted. Our Simulated Annealing (SA) cooling scheme implementation, with some parameter tuning, gave better results than our other methods, and more consistent solutions than the original method for some instances. Furthermore, to create other instances more easily, a form was created.

Keywords: High school, High school timetabling problem, meta-heuristics, Iterated Local Search, Torque operator, instance creator.

Contents

List of Tables

List of Figures

Introduction

Motivation and Context

Timetabling consists in assigning a set of activities to resources under complex constraints that vary depending of the problem context. It is an active area of research that has applications on universities, traffic, schools, hospitals, sports, business, etc. Furthermore, this problem is NP-Complete which cannot be solved in polynomial time by deterministic algorithms.

The *High School TimeTabling Problem* (HSTTP) is confronted in a lot of educational institutes worldwide. It consists in assigning resources such as teachers, and students in different time slots that represent lessons. However, this assignment has to satisfy a number of constraints such as avoiding the same teachers or students assist two lectures at the same time. The objective of this problem is to get a schedule that minimizes these conflicts.

These conflicts are divided into hard constraints, and soft constraints: the former must be satisfied since it represents the feasibility of the solution, whereas the latter represent preferences, and some of them might not be satisfied. A schedule is better if it violates less soft constraints.

Since there are many high schools, they might have different constraints, and some of them might be harder to satisfy. However, we assume that the corresponding high schools have already defined their constraints, and they have also defined which students attend which lessons, and the teachers who teach them. Furthermore, out of all these approaches, the one provided by Saviniec and Constantino (2017) has been selected to be improved since we believe we can improve it by setting a better perturbation operator.

There are no deterministic algorithms that can solve NP-Complete problems in polynomial time. However, according to Templatetypedef (2014), other kind of solutions can be applied such as: approximation algorithms, pseudopolynominal-time algorithms, randomized algorithms, parametrized algorithms, fast exponential-time algorithms and heuristics. The last ones are divided in heuristics, meta-heuristics, and hyper-heuristics, and, these last two have showed to solve these kind of problems efficiently. Furthermore, these are preferred since no specific knowledge is required for these problems, and they try not to get stuck in a local optimum. Out of these two options, meta-heuristics algorithms have been chosen to be implemented since hyper-heuristics just use a group of meta-heuristics, and they cannot be used if there are no meta-heuristics implemented beforehand. Moreover, hyper-heuristics' difficulty is just about tuning how often they will use meta-heuristics to solve a problem, and it is not the implementation itself (Herbawi, 2014).

From my point of view, better solutions to this problem will result in better high school schedules. Therefore, students' performance will increase, and this increased performance will help them perform better in college and other aspects of life.

Problem Statement

The problem being solved is the High School Timetabling Problem (HSTTP): given a set of requirements that provides information about the structure of each lesson, and a set of constraints with their respective weights, a feasible schedule that violates the least constraints is generated.

Objectives

To propose methods based on the Iterated Local Search - Torque (ILS-TQ) technique to solve the HSTTP.

Specific Objectives

- To research recent techniques in solving the HSTTP.
- To understand their operators, and how they modify their solutions.
- To implement the ILS-TQ technique by Saviniec and Constantino (2017).
- To propose methods to improve the ILS-TQ.
- To implement a form to create other instances to this problem for our methods.
- To compare results to the original ILS-TQ technique.

Thesis Organization

This work is organized as follows:

In Chapter 1, the definition of the specific HSTTP is given.

In Chapter 2 the state of the art of methods that try to solve variations of the HSTTP is given. These methods are focused on meta-heuristics.

In Chapter 3 the methods proposed to improve the ILS-TQ are explained in detail. Additional to our proposal, a form to create instances to this problem is shown too. At the end of this chapter, there is a section focused on explaining the difference between the original method, and ours.

In Chapter 4, all the results obtained by our methods are shown, they are also compared with the original method. At the end of this chapter, in order to compare how each method's schedules change over time, schedules given by these methods at the $10th$, $50th$, $100th$ and final iteration are shown.

Chapter 1

High School Timetabling Problem

The High School Timetabling Problem (HSTTP) consists in efficient assignation of two resources: teachers and classrooms through time. This assignment problem is more difficult due to some constraints such as teachers' preferences for scheduling and classrooms' availability.

The HSTTP that is being solved is the same as the problem by Saviniec and Constantino (2017) which only consists of five hard constraints, and three soft constraints, these hard constraints represent the feasibility of the solution (i.e. determines whether the schedule is impossible due to having the same teacher teach two classes at the same time), whereas the soft constrains represent the quality of the solution (i.e. the schedule is possible, but might be more or less preferable). Requirements, and teacher unavailable periods are represented by tuples. The following subsections explain these variables in more detail, but this explanation is very similar to the one by Saviniec and Constantino (2017) since it is the same problem.

1.1 Constraints

They also take the following constraints into consideration:

- Hard constraints: they represent the feasibility of the solution.
  - hc_1: each requirement must be assigned to exactly θ times a week.
  - hc_2: a class must attend exactly one meeting per period.
  - hc_3: a teacher must teach at most one lesson per period.
  - hc_4: teachers must not be assigned periods in which they are unavailable.
  - hc_5: each requirement must have less or equal than γ assignments per day.
- Soft constraints: they represent the quality of the solution.
  - sc_1: each requirement should have at least μ double lessons a week.

- sc_2: idle periods in the schedule of teachers should be avoided.
- sc_3: the teachers' schedules should be concentrated on a minimum number of days.

1.2 Timetabling Encoding

The input data to construct a timetable is given by two parameters:

- R: a set of tuples $(c \in C, t \in T, \theta \in N, \gamma \in N, \mu \in N)$ in which the subject of a class c is taught by a teacher t, and that subject has a duration of θ timeslots at most γ times per day. Also, that subject has a minimum number of weekly double lessons μ.
- U: teacher's unavailable periods are represented by a tuple $u \in U$ in the format $(t \in T, d \in D, h \in H)$ in which teacher t is unavailable at period h of day d.

The set of timeslots will be defined by P as $(d, h) \in D \times H$ where d is day, and h is hour. Timetables are represented by a 2D array Z in which the rows represent classes, and the columns represent timeslots. Each array's cell Z_{ej} points to a requirement tuple $r \in R$ assigned to class $c \in C$ in timeslot $j \in P$.

1.3 Objective Function

The objective function indicates how good a solution is, and it takes into consideration all soft constraints, and hard constraints, except for the first two of the latter since they are always satisfied.

Let β_i^{hc}, and β_i^{sc} be the number of times that constraint types $hc_i(i = 3, 4, 5)$, and $sc_i(i = 1, 2, 3)$ are violated, and let α_i^{hc}, and α_i^{sc} be the penalty constants associated to their respective violations to penalize them. Therefore, the objective function minimizes the following weighted sum:

$$\min f(Z) = \sum_{i=3}^{5} \alpha_i^{hc} \cdot \beta_i^{hc} + \sum_{i=1}^{3} \alpha_i^{sc} \cdot \beta_i^{sc} \tag{1.1}$$

The first term of (1.1) measures the feasibility of the solution which is represented by all the hard constraint violations whereas the second term measures the quality of the solution which is represented by all the soft constraint violations. Since the feasibility of the solution is more important than the quality of it, then all values of α_i^{hc} must be greater than α_i^{sc} to ensure that hard constraints are satisfied.

The values of β_i^{hc}, and β_i^{sc} are computed as follows:

$\beta_3^{hc} = \sum_{t \in T} \sum_{j=1}^{|P|} (\pi_{ij} - 1), \forall (\pi_{ij} > 1)$, where π_{ij} is the number of times teacher t has been assigned to a timeslot j.

$\beta_4^{hc} = \sum_{t \in T} \sum_{j=1}^{|P|} \rho_{ij}$, where $\rho_{ij} = 1$ if teacher t is assigned to teach at an unavailable timeslot j.

$\beta_5^{hc} = \sum_{r=1}^{|R|} \sum_{d \in D} (\sigma_{rd} - \gamma_r), \forall (\sigma_{rd} > \gamma_r)$, where σ_{rd} is the number of times the $r-th$ requirement has been assigned in a day, whereas γ_r is the maximum number of daily meetings.

$\beta_1^{sc} = \sum_{r=1}^{|R|} (\mu_r - \phi_r), \forall (\mu_r > \phi_r)$, where μ_r is the minimum number of double lessons for the $r - th$ requirement, and ϕ_r is the total number effectively scheduled.

$\beta_2^{sc} = \sum_{t \in T} \sum_{d \in D} \eta_{td}$, where η_{td} is the number of idle periods occur in the schedule of teacher t on day d.

$\beta_3^{sc} = \sum_{t \in T} x_t$, where x_t is the number of working days are scheduled to teacher t.

In their implementation, they used the following values for the corresponding penalties: $\alpha_3^{hc} = 100.000, \alpha_4^{hc} = 100.000, \alpha_5^{hc} = 10.000, \alpha_1^{sc} = 1, \alpha_2^{sc} = 3$, and $\alpha_3^{sc} = 9$

1.4 Final Consideration

The HSTTP problem we are aiming to solve has many variables such as constraints, classes, days, periods, requirements that must be taken into consideration in order to get a high quality schedule. However, this cannot be done with an exact method since there are many possibilities to build a schedule. There are q requirements that must be filled in $d \times t$ periods where d is the number of days, and t is the number of timeslots for each day, and each requirement can appear more than once. So, there are: $(d \times t)!$ permutations of these requirements for one class, and since there are c classes, it becomes: $(d \times t)!^c$ possible schedules that must be checked. Each schedule must be checked in order to know how good of a solution it is. Each objective function iterates over $(d \times t) \times c$ elements per schedule, so, its computational complexity is:

$$O(((d \times t)!^c) \times ((d \times t) \times c)) \tag{1.2}$$

For example, the smallest instance in our dataset has 4 classes, 5 days, and 5 timeslots, so there are: $100 \times (25!^4)$ different ways to form a schedule. 25! is approximately 10^{25} so $(25!)^4$ is approximately 10^{100}. Thus, there are roughly 10^{102} different schedules. A computer that does 10^9 operations per second would take 3×10^{85} years to compute all schedules. This shows that exact methods cannot be applied to these problems, so, meta-heuristics must be used in order to find feasible solutions in a reasonable time that is generally 10 minutes up to a day.

When the solution space is too large, exact methods cannot find the optimum solution in a reasonable time, so, meta-heuristics are generally used to find a near-optimum solution. In the next section, some methods that apply meta-heuristics to solve this problem are explored.

Chapter 2

State of the Art

The HSTTP has been studied for many years, and many solutions to its variations have been proposed throughout the years. However, since it has been proven to be an NP-Complete problem, exact solutions cannot be applied to real life high schools. Therefore, meta-heuristics have been used to solve bigger, and more complex instances. One problem with these techniques is that they cannot always be compared since they might generate a schedule for a specific school. The Kingston High school timetabling Engine (KHE) by Kingston (2015), which is an open-source ANSI C library which provides a fast, and robust foundation for solving problem instances related to the high school timetabling problem, is often used by some authors such as Brito, Fonseca, Toffolo, Haroldo, and Marcone (2012); Demirović and Musliu (2017); Fonseca and Santos (2014); Fonseca, Santos, and Carrano (2016); Fonseca, Santos, Toffolo, Brito, and Souza (2012, 2016); Yousef, Khader, Kheiri, and Ozcan (2013) to generate initial solutions. Meta-heuristics are methods used to find near-optimal solutions by perturbing a solution or group of solutions and storing the best found so far.

2.1 Iterated Local Search (ILS)

Gendreau and Potvin (2005) explain that Iterated Local Search (ILS) is a meta-heuristic that starts with a basic solution, and then finds the local optimum based on that solution. In order to escape this local optimum, it applies a perturbation operator to the solution, and then finds the local optimum of that solution.

Saviniec and Constantino (2017) propose a simple, yet efficient algorithm which is the classic ILS meta-heuristic but with a perturbation operator called Torque (TQ). It surpasses the results of Fonseca, Santos, Toffolo, et al. (2016) in almost all problem instances. They also propose another perturbation operator called Matching (MT) without significant improvements. It is similar to their earlier work by Saviniec, Aparecido, and Romao (2013), but with a different local search.

Their TQ operator constructs a graph by choosing two periods t_i, t_j. Each vertex in the graph represents the pair (t_i, t_j) for each class, and they are connected if, and only if there is a teacher clash by swapping the pair of one vertex. The generated graph consists of one or more connected components, and each TQ move consists in swapping all the pair for each vertex that belongs to a component. Thus, a new schedule is generated.

The perturbation operator is a single TQ move where t_i, t_j, and the component to be swapped are chosen randomly. Their local search is done by iterating through all possible timeslot pairs, and for each pair, they generate its corresponding graph. Each of its components are swapped generating new solutions. If the new solution improves the current one, then former replaces the latter, and the remaining components are swapped. This process is repeated until there is no further improvement. At the end of the local search, the local optimum is returned. The new local optimum is compared with the best solution so far, and replaces it if it is better. If three local optimum cannot improve the best solution so far, the current solution is reset to the best solution so far. The stopping criterion is a time limit of ten minutes.

2.2 Variable Neighborhood Search (VNS)

The main characteristic of the Variable Neighborhood Search (VNS) is that it considers a number of neighborhoods to be explored, and each one is explored according to its success: if there is no better solution, then the next neighborhood is explored. If a better solution is found, it goes back to the first neighborhood Gendreau and Potvin (2005). A neighborhood is formed by all solutions that can be reached with a certain operator, and the closest neighborhood of a solution is made by all the solutions that can be reached by applying an operator once to that solution.

Saviniec and Constantino (2017) also propose a VNS which uses the MT, and TQ operators. However, no significant improvement was done compared to their ILS Implementation. The most important part of their implementation are both operators. The MT operator selects a set of requirements of the same class, builds an $N \times N$ cost matrix where N is the number of requirements with their respective timeslots, and each cell is filled with the cost of inserting the requirement i in the timeslot j. This cost matrix is used to solve the corresponding assignment problem by applying the primal–dual algorithm by Carpaneto and Toth (1987). The solution of the assignment problem is the permutation that minimizes the objective function.

The VNS-HS by Fonseca and Santos (2014) starts by using the KHE, and then applies the VNS with improvements on some instances. They compare some variants of this meta-heuristic: The Reduced Variable Neighborhood Search (RVNS) has no descent phase, therefore improving its computation time when the neighborhood structure is extensive; the Sequential Variable Neighborhood Descent (SVND) only explores the $k-th$ neighborhood at each iteration k; and the Skewed Variable Neighborhood Search (SVNS) also accepts worse candidate solutions by following a relaxed rule, that calculates the distance between the candidate solution, and the best solution so far, and that considers one neighborhood structure in each iteration. Since the last variation gives better results compared to the other two variations, some relaxed rules might improve Saviniec et al.'s Saviniec and Constantino (2017) method.

2.3 Tabu Search (TS)

Tabu Search (TS) is a meta-heuristic that moves from the current solution to the best neighbor that is not in the tabu list. This is used to avoid cycling moves Glover (1989).

The TS by Minh, Thanh, Trang, and Hue (2010) builds an initial solution with a greedy algorithm that splits each course in blocks (i.e. consecutive lectures) to be assigned on consecutive periods. Each course is split in different combinations of blocks (i.e. block splitting way) according to the constraints, then it calculates all period-assigning ways for each block for each course. After that, it selects the block-splitting way that has the block that belongs to the course with the smallest number of possible period-assigning ways to be placed in a period-assigning way. That causes the least reduction of those period-assigning ways. Their initial solution is improved with TS primarily by using single moves. It also applies swap moves, and block-changing moves when there is no improvement with single moves. There are two tabu lists: one for single moves, and another one for block-changing moves. They applied their algorithm to three real-world instances of two high schools of Vietnam, but there are no comparison with standard benchmarks. The initial solution constructive heuristic is more expensive than Saviniec and Constantino (2017) greedy algorithm, but it might provide a better initial solution.

2.4 Simulated Annealing (SA)

Simulated Annealing (SA) is a meta-heuristic that tries to find the global optimum in a large search space (Du & Swamy, 2016). It starts with an initial temperature which decreases with each iteration by a factor of α, and solutions that are worse may be accepted according to the current temperature, and the difference between the current solution, and the best one.

Zhang, Liu, M'Hallah, and Leung (2010) propose a SA to generate their initial solution, and another one to improve the initial solution. The first SA that they use selects a period that causes conflict in any day to exchange it with another random period for a random permutation of classes. It updates the best solution so far if the swap neighbor improves it. Their second SA generates a random permutation of periods where the first one is selected, and swapped with the rest of the periods in the permutation for each random class. It updates the best solution so far if the swap neighbor improves it, and it does not violate any hard constraints. They have two groups of datasets: the hdtt4-hdtt8, and from Greek schools. A modification to its local search might be used as a perturbation phase since it is not as expensive as Minh et al. (2010).

The SA proposed by Demirovic and Musliu (2017) uses a bitvector representation to improve an initial solution generated with the Kingston High school timetabling Engine (KHE). For each lesson, it creates $2 + m$ of vectors of length n where m is the number of different durations each sub-lesson has, and n is the number of total periods. The first vector has a 1 set in position i if the sub-lesson is taught at position i, the second vector has a 1 set in position i if a sub-lesson starts at that position, and the following vectors have a 1 set in position i if a sub-lesson of length d starts at that position, where d is the duration of that lesson, and each different d belongs to a vector. This representation is efficient to calculate the quality of each solution. However, their representation is best used as an initial solution generator since it performs better than KHE, but it does not give competitive results compared to the best existing solutions, but those solvers did not have time nor resources constraints. Furthermore, they use the XHSTT-2014 dataset, but they could model 23 out of 39 instances with bitvectors. Still, it might be worth representing the solution as bitvectors when finding an initial solution

Similarly, Brito, Fonseca, and Haroldo (2012) propose a technique that starts with an initial solution generated by KHE, and improve it with SA. It got the best results for some instances in the International Timetabling Competition (ITC) 2011 dataset.

Odeniyi, Omidiora, Olabiyisi, and Aluko (2015) modify the cooling schedule to become parabolic which takes less computation time, and less computational cost compared to the original SA. In other words, the original SA has the reduction parameter $\alpha = \frac{1}{\log 1+t}$ where t is the temperature, and they changed it to $\alpha = \frac{1}{\log 1+t+t^2}$. Their approach was used to generate "the first time school timetable of Fakunle Comprehensive High School, Osogbo Nigeria during the 2012/2013 session". Although it sounds promising, their modification is required to be tested in more instances to see whether it truly improves over the original SA.

2.5 Cat Swarm Optimization (CSO)

Cat Swarm Optimization (CSO) has a population of solutions represented as cats in which each cat behaves in two modes: seeking mode, and tracing mode. In every iteration, cats are sorted into these two modes where the quality of the solutions are bound to change (Bahrami, Bozorg-Haddad, & Chu, 2006).

Skoullis, Tassopoulos, and Beligiannis (2016) propose a hybrid CSO for greek schools. Their proposal consists in a modified CSO with a local search at the end. In their seeking mode, copies of a cat are made, and for each copy the value of Count of Dimensions to Change (CDC) is changed at random. After calculating each solution's fitness, and each solution's probability to be chosen is changed, and, after selection, the cat is moved to that position. In their tracing mode, each cat updates its velocity according to the best cat's position, and it is set to the maximum velocity if the new one exceeds it. The position of the cat is updated. Each cat is represented as a 2D array where rows represent classes, and columns represent periods, similar to the Direct Genetic Algorithm (DGA) representation. They also have auxiliary procedures that involve swapping cells: the change_random(), which replaces a random whole column with the best cat's column, and swaps cells horizontally if there is a conflict, and the single_swap(), which swaps two cells only if they have different values, they are not empty, and if they do not produce a teacher clash. Furthermore, their method achieves better results compared to the majority of algorithms when the same instances are used: ten widely used school timetabling instances where six of these constitute the Beligiannis benchmark. This technique finds comparable results to the Hybrid Particle Swarm Optimization (PSO) by Katsaragakis, Tassopoulos, and Beligiannis (2015), but takes less execution time , and even though CSO consists of many solutions, the auxiliary operators might be worth a try as part of the perturbation phase.

2.6 Genetic Algorithms (GA)

This meta-heuristic starts with a solution population that can be randomly generated. Then, it selects a number of solutions which can also be repeated, and then it applies a crossover operation to each other. A mutation operator is randomly applied to the new population to escape the local optimum, and this process is repeated over the new population (Carr, 2014).

The Genetic Algorithm (GA) by Febrita and Mahmudy (2017) combines GA Fuzzy Time Window to solve instances at a private school at Malang. The Fuzzy Window affects the quality of the solutions. Subjects are divided in exact, and non-exact subjects in which the exact subjects need to be scheduled between the first, and fourth period, and the non-exact subjects can be placed at the remaining periods. However, if an exact subject is not scheduled within those periods, then its satisfaction level is less than one. Solutions are represented as a 2D matrix where rows are the periods, and columns represent the classes, and each gene (i.e. cell) is represented by an integer. Its crossover phase consists in swapping two sub-matrices of two parents to generate two children. In the mutation phase, two points are selected for two random classes, the first point is a gene with a unsatisfied time window while the other is a non-exact subject. The selection operation used is the replacement selection mechanism. They used the dataset taken at a private school in Malang with five school days, and seven periods per day. More tests need to be made, but having a Fuzzy Time Window affect mutations, and fitness sounds interesting. However, the dataset we used does not distinguish between exact, and non-exact subjects.

Saptarini, Suasnawa, and Ciptayani (2018) apply a direct representation GA to groups of solutions where each group computed a different GA, and their solutions might migrate to other groups. Their solutions are represented as a 2D matrix where the rows represent time, and the columns represent the class' name. In this representation, each cell has an integer that represents the teacher, and subject (i.e. course); their selection is the roulette wheel method where the higher the quality of the solution, the more likely it is to be selected for the next generation of solutions; each solution is mutated based on two operators: change, and swap where the former is used when a teacher teaches less times than the other teachers in the same field, whereas the latter is done otherwise; their crossover is a one-point crossover for being simple. Elitism is also implemented. Their most important part is the migration step that helps the algorithm not to get stuck at a local optimum, this is done by having a migration probability that decides whether to choose a random solution to migrate to another group. This algorithm was applied to a specific case, so more tests must be done to actually say for certain whether it is a viable solver or not; regardless of that, their mutation operators are common, and seem less likely to improve the method by Saviniec and Constantino (2017)

Raghavjee and Pillay (2013) compare the two-phase Direct, and Indirect GA to solve the African HSTTP; the former consists of solutions represented by a matrix whereas the latter consists of solutions represented by a string of instructions. It's called two-phase because the first phase generates a feasible solution while the second phase improves it. In the DGA, each solution is represented as a 2D matrix similar to what Saptarini et al. (2018) did, but each cell stores a tuple that consists of a teacher, its subject, and the venue if applicable. It uses the tournament selection in phase one, and both the original, and modified one in phase two. It has four mutation operators for phase one, and they are all swaps, but they differ whether the swap results in a better solution or not, and whether one or two constraint violating tuple are chosen for the swap, whereas it also has four mutation operators for phase two in which one is a random swap, the other one is a row swap, the third one, and the fourth one as the two operators for phase one that select one or two tuples that violate constraints. All these operators for phase two consider improvement over the last solution. For the Indirect Genetic Algorithm (IGA) representation, each solution is represented as a string of instructions where each one is represented as a character (A, D, 1..4, 5..8 for Allocation of a random tuple, Deallocation of a random tuple, operators for phase one, operators for phase two respectively). These instructions are capable of building or modifying a timetable. Each solution is generated randomly, but it is possible that the string might not contain all possible tuples. Their method uses the cut-and-splice crossover, and the unmodified tournament selection method. Unfortunately, both algorithms were tested on only two instances, but that IGA performed far better than DGA. So, it might be beneficial to change our timetabling encoding to a string of instructions, but that would change many things in the current implementation.

2.7 Hybrid Meta-heuristics

Hybrid solutions consist in combining two or more meta-heuristics. By combination, it means that one meta-heuristic is used after the other or it is used within another one.

Fonseca et al. (2012) approach consists in first generating a solution by using the KHE. Then, they apply a SA, and finally, they apply an ILS. Two years later, they improve their solution, called GOAL Fonseca, Santos, Toffolo, et al. (2016), by changing which neighborhoods to use in the perturbation phase in their Iterated Local Search (ILS) implementation

Fonseca, Santos, and Carrano (2016) also generate their solution using the KHE. Then, they apply SA to improve the solution (similar to Fonseca, Santos, Toffolo, et al. (2016)). The difference is in the last step where they apply the late acceptance stagnation-free Hill climbing heuristic that stores the best solutions in an array. Their meta-heuristic is created in order not to have an artificial cooling schedule, to use information of previous iterations of the search, and to have a simple acceptance mechanism. However, the difference with this variation of the acceptance Hill climbing heuristic is that the array is restored to the last improvement if there is no improvement after a certain number of iterations. The results of this hybrid approach have also improved over the results by Fonseca, Santos, Toffolo, et al. (2016) in 15 instances out of 18 in the ITC Post, Gaspero, Kingston, McCollum, and Schaerf (2016) 2011 dataset. The most important, and promising part is the heuristic they use which is simple to implement, it has only one parameter that is the size of the array, and it does not add the need to change the implementation by Saviniec and Constantino (2017). Furthermore, it provided better results in a number of standard instances over the method by Fonseca, Santos, Toffolo, et al. (2016).

The technique by Brito, Fonseca, Toffolo, et al. (2012) generates an initial solution with KHE too, and then improves it with SA, and VNS afterwards. Moreover, they also implement two variations of VNS which are the RVNS, and General Variable Neighborhood Search (GVNS): The RVNS variation is obtained when no local search is made, and just random solutions are obtained from the neighborhood, whereas the GVNS is achieved when the local search is replaced by a Variable Neighborhood Descent (VND) in which the change of neighborhoods is done in a deterministic way. The GVNS variation performs better for small, and medium-sized instances whereas the RVNS performs better for large instances since the VND method takes more time as the size increases. They also run their implementations on the ITC 2011 dataset. Depending on the size of our instance, a VND might prove more beneficial than the current local search for the method by Saviniec and Constantino (2017).

Demirović and Musliu (2017) also generate a solution using the KHE (KHE14) or by ignoring the soft constraints, then they improve it with a local search based on SA, and finally they improve it even further with Large Neighborhood Search (LNS) by unscheduling, and rescheduling subevents while recording each operator's performance. Each solution is encoded as a Partial Weighted maxSAT problem in which clauses are partitioned in hard, and soft clauses, and each soft clause has a weight. The objective is to satisfy the hard constraints, and minimize the accumulated weight of all soft clauses violated, similar to the HSTTP. Boolean variables $Y_{e,t}$ represent each maxSAT representation where e is an event, and t is the timeslot it is taking place, so, a solution consists in assigning truth values to each of these variables. The local search based on SA uses two neighborhoods that are swap, and block-swap. After that, a LNS is used to find a near optimum solution, it consists of two operators that destroy, and insert the solution. The destroy operator consists of two neighborhood vectors based on resources, and based on days where all possible combination of resource-pair, and days are inserted in their corresponding vectors, and they are ordered randomly when a vector becomes active; only one vector becomes active after a timeout or if all the neighborhoods have been visited twice. As for the insert operator, it finds the best possible insertion for the unscheduled events by searching exhaustively on the maxSAT formulation. Trying to adapt the method by

Saviniec and Constantino (2017) to this implementation would require a lot of work, so it will not be our top priority.

Fonseca and Santos (2013) use a memetic algorithm which is similar to GA, but with a refinement phase where they apply SA, and then ILS to it, they start with initial solutions generated by KHE. For the crossover phase, their method splits the population in two, and selects the $i-th$ solution in both groups so that they become the new parents based on the crossover rate; if they do, then each parent produces a exact copy of itself, and for each cell, according to a probability, both parent's cells are swapped, and this is reflected in both children analogously. Each solution is mutated by a Lesson Swap or a Resource Swap. For the selection phase, it runs a tournament selection where elitism is implemented. For the refinement phase, as said before, SA, and ILS are applied to each solution. Their technique overcomes the method by Fonseca, Santos, Toffolo, et al. (2016) using the ITC 2011 dataset. Since Saviniec and Constantino (2017) is not a GA, then this method is not that useful. Furthermore, their operator is not new, so it might not improve our modification.

Similarly, Yousef et al. (2013) also begin with solutions by KHE which are improved by applying Harmony Search (HS) which is a population meta-heuristic that replaces the worst solution with the new one if the latter is better, and then SA. Each solution is represented as an array which is called a harmony vector. Since the hard constraints of the initial solutions are likely to be violated, the rest of the algorithm needs to fix that. New harmony vectors are generated by either copying each element from an already existing vector or by applying a neighborhood move to it so that the new vector will have some elements from its parent. It has three neighborhood moves which are Move Meeting, Swap Meeting, and Do Nothing, and all three have equal probability. If this new vector has better quality than the worst harmony vector, then the worst one is replaced by the new one. After each iteration, SA is applied in order to improve the best solution obtained, and five neighborhoods are considered: Move Meeting, Swap Meeting, Swap Three Meetings, Swap Block of Meetings, and Task Split. They use the ITC 2011 dataset, but did not improve any best solution found by other methods. The Swap Three Meetings neighborhood seems interesting, and not hard to implement compared to the classical operators such as Move meeting, and Swap Meeting that have already been used many times, Swap Block of Meetings does not apply to the ILS-TQ swaps since only two cells are swapped, and we cannot split our tasks for the last neighborhood.

Sutar and Bichkar (2017) initialize a solution with workload fulfillment, then improve it with TS by swapping room, teacher pairs. After that, they keep the best solutions in an array for their GA which prevents adding other clashes. This is done with their mutation operator that is more likely to be applied the more errors the solution has, it checks for room/teacher clashes, and it replaces the clashing value with a random number in the range of room/teacher values. Furthermore, they also implemented an intelligent crossover that maintains workload feasibility. Their TS only has random room, teacher pair swaps applied a number of times to each solution, and the best one of these swaps is chosen, the pairs' indices are stored in the tabu list to avoid these moves. Finally, the best solution of the last iteration is used to initialize GA. Their technique converges, and gives solutions within a few seconds for the hdtt4, ”hard timetabling” dataset in OR-library. The most interesting part that can be implemented is the mutation part, the crossover phase is interesting too, but the ILS-TQ only works with a single solution.

2.8 Final Considerations

The main problem with these methods is that they do not always use standard datasets, so it is not possible to directly compare them. We chose to improve the ILS-TQ technique because it improved over GOAL's results, and it might show improvement if its perturbation operator is modified. So, what we can do is implement some perturbation/mutation operators from the methods explained above. These operators are modified versions of Fonseca and Santos (2014); Skoullis et al. (2016); Yousef et al. (2013); Zhang et al. (2010), a two-move TQ and a hybrid method; they have been chosen because of their simplicity, how well we can adapt them to the ILS-TQ, and the results they obtained.

Chapter 3

Modified ILS-TQ

Our proposal are modifications to the original by Saviniec and Constantino (2017). Their method provides solutions with no hard constraints to 34 real-case instances that are from 2008, 2010, and 2011 which were collected from thirteen schools in Brazil. Furthermore, their method improved upon GOAL's results. All the pseudo-codes shown in this chapter are from Saviniec and Constantino (2017).

Figure 3.1 shows an overview of all our modifications: modifications pertaining to how a solution is accepted (schedule acceptance criterion) are painted blue, modifications pertaining to how a schedule is perturbed are painted red and the rest is painted purple which is both modifications in one.

3.1 Input Encoding

Requirements are represented as a $C \times T$ matrix where C is the number of classes, and T is the number of teachers. Each cell has a three-tuple (θ, γ, μ) that belongs to the requirement that has c as a class, and t as a teacher, θ as the duration in timeslots, γ as times per day, and μ as the minimum number of weekly double lessons. Requirements are represented that way since the pair (c, t) for each requirement is unique for each instance. This is logically represented as a vector of length $C \times T$. Although there are empty cells in this representation, access to each requirement is fast since it is all in a contiguous memory block. Empty cells are represented as $(0, 0, 0)$.

Teacher unavailabilities are represented as a $T \times P$ matrix where T is the total number of teachers and P is the total number of periods. Each cell has a value of 1 or 0 where 1 means the teacher t is unavailable at period p, and 0 means otherwise. This is logically represented as a vector of length $T \times P$. Access to each position (t, p) is fast since all values are in a contiguous memory block.

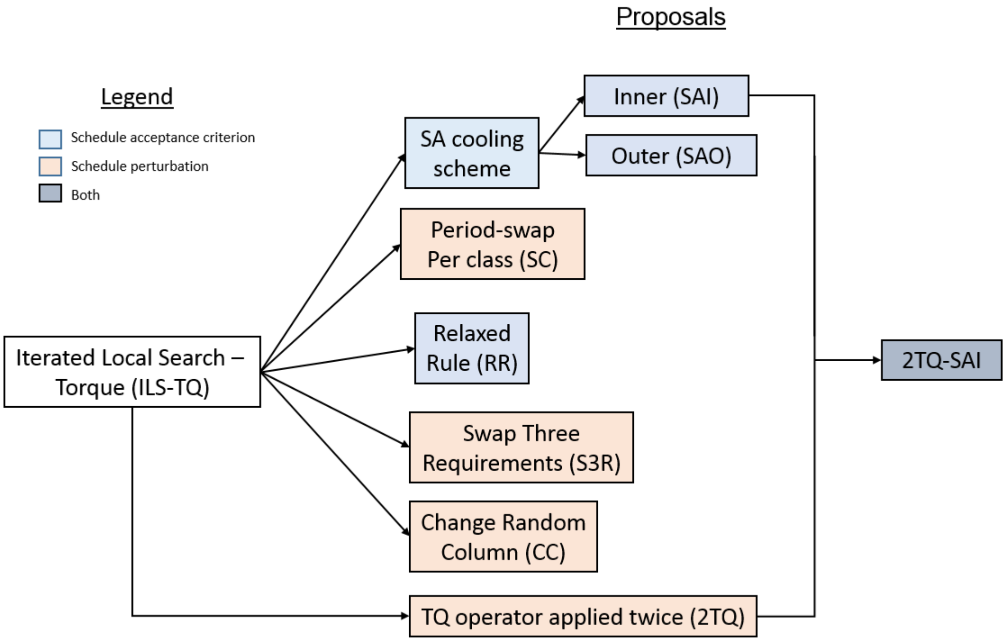


Figure 3.1: Overview of all our modifications. Blue modifications change how a solution is accepted; red modifications change how a solution is perturbed and the purple modification combines both.

3.2 Constraint Violation Calculation

The calculation of all objective functions was done by using the following data structures:

- **Teacher_timeslots**: a $T \times P$ matrix that stores how many times teacher t teaches in each period p. This is used to calculate violations to the third hard constraint: a teacher must teach at most one lesson per period. This is logically represented as a vector.
- **Assignments**: an $R \times D$ matrix where R is the number of requirements, and D is the number of days. It stores the number of times a requirement r is taught in a day. This is used to calculate violations to the fifth hard constraint: each requirement must have less or equal than γ assignments per day. This is logically represented as a vector.
- **Teacher_wd**: a $T \times D$ matrix that stores which days teacher t teaches in each week. This is used to calculate violations to the third soft constraint: the teachers' schedules should be concentrated on a minimum number of days. This is logically represented as a vector.
- **Double_lessons**: a R vector that stores the number of double lessons a requirement r has per week. This is used to calculate violations to the first soft constraint: each requirement should have at least μ double lessons a week.
- **Unavbls**: a $T \times P$ matrix that stores the teacher t that is unavailable at period p where 1 means that the teacher is unavailable at that period, and 0 means otherwise.

This is used to calculate violations to the fourth hard constraint: teachers must not be assigned periods in which they are unavailable. This is logically represented as a vector.

- **Idle_tbl**: a 2^{DP} vector where DP is a number of periods per day. It stores all combinations of idle times in a day. This is used to calculate violations to the second soft constraint: idle periods in the schedule of teachers should be avoided. While this data structure is efficient, it does not allow instances with more than 5 days or 5 periods per day to be solved since this vector is hard coded.

3.3 Torque Operator (TQ)

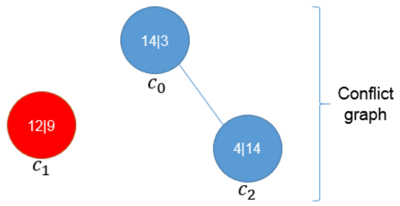
The Torque (TQ) operator builds a conflict graph in which each vertex represents a pair of different requirements $r_i, r_j \in R$ for a class, and each edge represents a conflict between this pair, and another pair of requirements from another class when two requirements of a class are swapped. After all connected components are identified, each component represents a chain of one or more swap moves that generates a new neighboring solution that minimizes conflicts.

Figure 3.2 shows how the TQ operator is applied to a schedule, that shows teachers' ids, to build a conflict graph. Two random timeslots (t_9 and t_{15}) are selected. If we swap those timeslots for class c_0, there is a teacher clash with c_2 (i.e. teacher 14 teaches on timeslot t_{15} at the same time on c_0 and c_2). So, they are connected in the conflict graph. There is no conflict with c_2 and the rest of the classes, so it is another separate connected component. In the end, there are two connected components, the blue one is chosen, so all pairs of requirements in that connected component are swapped, and the new schedule is a neighbor of the current schedule.

3.4 Constructive Algorithm

Initial solutions are constructed by the randomized heuristic shown in Algorithm 1 by Saviniec and Constantino (2017) that receives a set of requirements R, and for each requirement e that belongs to R, e's address is stored in $p(e)$, its class is stored in c' and the number of lessons for that requirement is stored in $numLessons$. Finally, $p(e)$ is randomly placed in $numLessons$ empty timeslots for class c'.
Solutions represented this way satisfy the constraints hc_1 (i.e. each requirement must be assigned to exactly θ times a week), and hc_2 (i.e. a class must attend exactly one meeting per period). To sum up, the algorithm iterates through all requirements, and randomly assigns them to empty timeslots according to their class.

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	14	2	10	4	13	2	3	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6



Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	3	2	10	4	13	2	14	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	7	5	3	6	8	4	0	14	7	4	3	9	1	0	6

Figure 3.2: TQ Move applied to a schedule with teachers' ids, two random timeslots (t_9 and t_{15}) are selected. Conflicts are highlighted in blue and red in the conflict graph. The blue connected component is chosen and its requirements are swapped.

Algorithm 1 Pseudo-code of the constructive algorithm by Saviniec and Constantino (2017)

function CONSTRUCT-SOLUTION(R)
  Initialize an empty solution Z.
  for each $e \in R$ **do**
    Let $p(e)$ be a pointer to e.
    $c' = e.c$
    $numLessons = e.\theta$
    while $numLessons > 0$ **do**
      Let $P_{c'}$ be the subset of timeslots $j \in P$ for which $Z_{c'j}$ is empty to class c'.
      Choose a random timeslot $j \in P_{c'}$
      $Z_{c'j} = p(e)$
      numLessons = numLessons - 1
    end while
  end for
  return Z
end function

3.5 Iterated Local Search

The local search as seen in Algorithm 2 by Saviniec and Constantino (2017) receives a solution Z as a parameter, sets P as the set of timeslots in the solution, and CC as the connected components in a graph G of conflicting meetings. The score of the solution Z is stored in f', and for every unique timeslot pair (i, j), it constructs the conflict graph G of solution Z and updates CC to store the connected components of G. For every connected component k in CC, all the requirements in k are swapped to obtain a new solution Z'' that will replace the current solution Z if it is not worse. The algorithm stops until no further improvement is made compared to f'. To sum it up, it builds a new conflict graph for every period pair, and for every component of the graph, it swaps all nodes that belong to that component, and the new solution is changed to be the best one if it is not worse than the current one. It stops until no more improvement is made.

Algorithm 2 Pseudo-code of the local search with the TQ operator by Saviniec and Constantino (2017)

```
function LocalSearchTQ(Z)
    Let P be the set of timeslots defined in Section 4.1.
    Let CC be the set of connected components in a graph G of conflicting meetings.
    do
        f′ = f(Z)
        for each (i, j ∈ P; i ≠ j) do
            Construct the graph G for requirements assigned to timeslots i, and j of solution Z.
            Compute the connected components of G, and update CC.
            for each (k ∈ CC) do
                Swap the requirements at the component k to obtain a neighboring solution Z″.
                if f(Z″) ≤ f(Z) then
                    Z = Z″
                end if
            end for
        end for
    while f(Z) < f′
    return Z
end function
```

The implementation of Algorithm 3 by Saviniec and Constantino (2017) is similar to the classic ILS: it starts with a solution Z and the number of seconds stored in t_{max}, the best solution Z^* is initially the current solution Z. The solution is perturbed by applying one TQ move, and then it is local searched to improve the solution Z, if this solution is strictly better than the best one found so far, then the counter $NotImproved$ goes back to 0. Otherwise, it is incremented. The best solution is stored in Z^* if the current solution not worse than the best one. If there was no improvement in the last 3 iterations, the counter is reset and the current solution is the best one. Here, the modification to the classic ILS is given by the number of times the current solution shows no improvement: if it is equal or greater than three, then it resets the current solution to the best one found so far. The implementation runs for t_{max} seconds.

Algorithm 3 Pseudo-code of the ILS-TQ algorithm by Saviniec and Constantino (2017)

```
function ILS-TQ(Z, t_max)
    Z* = Z.
    NotImproved = 0.
    while CpuTime() < t_max do
        Z = Perturbation(Z, 1)
        Z = LocalSearchTQ(Z)
        if f(Z) < f(Z*) then
            NotImproved = 0
        else
            NotImproved = NotImproved + 1
        end if
        if f(Z) ≤ f(Z*) then
            Z* = Z
        end if
        if NotImproved ≥ 3 then
            Z = Z*
            NotImproved = 0
        end if
    end while
    return Z*
end function
```

3.6 Modifications

The perturbation phase of Saviniec et al.'s method consists of a single TQ move that randomly chooses between two periods. We believe that a modification in this perturbation phase will help improve its results if it is explored more. Furthermore, applying relaxed rules might help find a better solution.

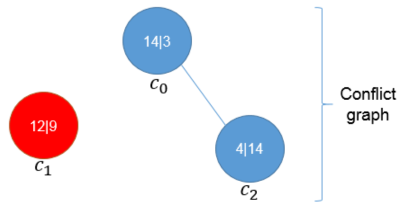
3.6.1 Two ILS-TQ (2TQ)

The problem with a single random TQ move is that it might get canceled in the local search phase with another TQ move. So, in order for this not to happen, two TQ moves are applied instead of one.

Figure 3.3 shows an example of this method applied to a schedule with teachers' ids. The timeslots are randomly selected, the second conflict graph is built after one connected component from the first graph has been swapped. The new schedule after the second TQ move is the new neighbor. The blue component from the first conflict graph has been chosen, and the green component from the second conflict graph was chosen.

a)

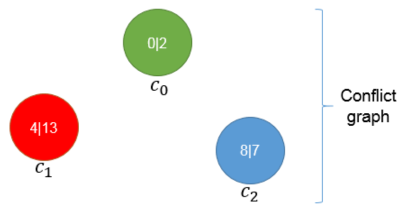
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	14	2	10	4	13	2	3	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6



Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	3	2	10	4	13	2	14	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	7	5	3	6	8	4	0	14	7	4	3	9	1	0	6

b)

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	3	2	10	4	13	2	14	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	7	5	3	6	8	4	0	14	7	4	3	9	1	0	6



Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	2	6	11	13	9	3	0	10	4	13	2	14	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	7	5	3	6	8	4	0	14	7	4	3	9	1	0	6

Figure 3.3: 2TQ applied to two schedules with teachers' ids. a) two random timeslots (t_9 and t_{15}) are selected and the requirements in the blue connected component are swapped. b) Two random timeslots (t_4 and t_{10}) are selected and the components in the green connected component are swapped. Conflicts are highlighted in blue, red and green in the conflict graphs.

	Teachers																								
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	3	3	4	13	13	6	4	4	9	9	14	14	0	11	11
c_1	7	7	6	13	13	7	5	5	7	12	4	4	3	8	8	9	9	6	12	4	0	0	3	14	14
c_2	6	13	7	7	8	9	7	7	5	5	6	13	8	4	4	1	1	9	4	14	3	3	14	0	0

Best schedule found so far (S)

	Teachers																								
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	2	0	2	3	3	4	13	13	6	4	4	9	9	14	14	0	11	11
c_1	7	7	6	13	13	7	5	5	7	12	4	4	3	8	8	9	9	6	12	4	0	0	3	14	14
c_2	6	13	7	7	8	9	7	7	5	5	6	13	8	4	4	1	1	9	4	14	3	3	14	0	0

Current Schedule (S')

$$\underbrace{f'(S')}_{\text{New score of S'}} = \underbrace{f(S')}_{\text{Current score of S'}} - \underset{\text{Constant = 1.0}}{\alpha} * \underbrace{\rho(S,S')}_{\text{Distance between S and S'}}$$

Figure 3.4: RR applied to one schedule with teachers' ids. The upper schedule S is the best found so far, and the lower schedule S' is the current one after a local search. Its new score is only to compare against the best solution found.

3.6.2 Relaxed Rule (RR)

The original ILS-TQ has no relaxed rules, meaning that it will not accept solutions that are worse than the best one. This is not a perturbation operator per se, but it will allow more solutions to be explored. This is a rule used by Fonseca and Santos (2014) that improved over their variations. In other words, the evaluation function of the new solution $f(S")$ is replaced by $f(S") - \alpha \times \rho(S, S")$ where $S"$ is the new solution, S is the best solution, α is 1.0, and $\rho(S, S")$ is the distance between the two solutions (i.e. the number of cells between the two solutions are different). This relaxed rule is placed at Line 11 in Algorithm 2

Figure 3.4 shows this relaxed rule applied to a schedule S' that has been found after a local search (i.e. lower schedule). The upper schedule S is the best schedule that has a score of $f(S)$, whereas the current schedule has a score of $f(S')$, its new score is only used to compare against the best solution found.

3.6.3 Random Swaps per Class (SC)

This perturbation consists in swapping two random periods for each class that is chosen randomly. This perturbation does not get canceled like the original perturbation phase. This is a modified version of the local search used in the second SA by Zhang et al. (2010).

Figure 3.5 shows an example of this method applied to a schedule with teachers' ids. Classes are selected in a random order and timeslots are randomly chosen. For c_0, timeslots t_4 and t_{16} are swapped; c_2, timeslots t_3 and t_7 are swapped; c_1, timeslots t_{19} and t_{20} are swapped.

Current Schedule

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	14	2	10	4	13	2	3	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Step 1

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	4	6	11	13	9	14	2	10	4	13	2	3	0	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Step 2

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	4	6	11	13	9	14	2	10	4	13	2	3	0	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	5	8	1	7	7	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Step 3

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	4	6	11	13	9	14	2	10	4	13	2	3	0	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	3	0	12	4	7	6
c_2	9	13	13	5	8	1	7	7	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Figure 3.5: SC applied to the current schedule with teachers' ids. In each step, a random new class is chosen, and two timeslots are randomly chosen to be swapped.

3.6.4 Change Random Column (CC)

This move copies a random $i-th$ column of the best solution to the current solution's $i-th$ column, and replaces the new solution's requirements with extra periods with those that lack periods. This is one operator used by the technique by Skoullis et al. (2016)

Figure 3.6 shows how this method is applied to the current schedule. The fifth column (i.e. timeslot t_4) was randomly chosen from the best schedule (upper one) to be copied to the current schedule. However, in order to copy this column, timeslots t_4 and t_{11} are chosen for class c_0, and timeslots t_4 and t_7 are chosen for class c_1 to be swapped so that the current solution's second column can be the same as the best schedule's.

3.6.5 Swap Three Requirements (S3R)

This operator swaps requirements r_1, and r_2, and then it swaps requirements r_2, and r_3 for each node in the components generated in the two conflict graphs for three random timeslots t_1, t_2, t_3. This is a modification of the operator by Yousef et al. (2013), but applying the TQ operator.

Random period chosen

Best schedule found so far

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	3	3	4	13	13	6	4	4	9	9	14	14	0	11	11
c_1	7	7	6	13	13	7	5	5	7	12	4	4	3	8	8	9	9	6	12	4	0	0	3	14	14
c_2	6	13	7	7	8	9	7	7	5	5	6	13	8	4	4	1	1	9	4	14	3	3	14	0	0

Current Schedule

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	14	2	10	4	13	2	3	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	14	2	10	4	13	2	3	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	10	~~6~~	~~11~~	~~13~~	~~9~~	~~14~~	~~2~~	0	4	13	2	3	4	3	1	0	14	11	6	0	8
c_1	14	5	7	9	13	~~7~~	~~0~~	4	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6

Figure 3.6: CC applied to the current schedule with teachers' ids with timeslot t_4 as the chosen column to copy from the best schedule. Timeslots t_{11} and t_7 are chosen for classes c_0 and c_1 respectively, and swapped with timeslot t_4. Finally, the column has been copied.

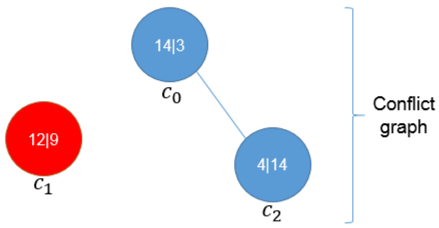
Figure 3.7 shows an example of this method applied to a current schedule. This is similar to the method 2TQ, except for the fact that the last timeslot chosen in the first TQ move is the first timeslot chosen in the second TQ move. In this case, timeslots t_9 and t_{15} have been chosen in this order for the first TQ move, and the blue connected component has been chosen. So, timeslots t_{15} and t_{21} have been chosen in this order for the second TQ move, and the blue connected component has also been chosen.

3.6.6 Simulated Annealing (SA) Cooling Scheme

We decided to implement the SA cooling scheme within our implementation of the Iterated Local Search Torque ILS-TQ. Similar to the method RR, this may also accept worse results, but according to the following criterion: $r < e^{\frac{f(S)-f(S")}{t}}$ modified from Zhang et al. (2010) where r is a random number between 0 and 1 inclusive, $f(S)$ is the score of the best solution, $f(S")$ is the score of the current solution and t is the temperature. The latter is a parameter that is tuned, but the temperature changes according to a cooling scheme given by $t = \alpha \times t$ where α is the cooling rate and this parameter is also tuned. The criterion is placed in the local search and evaluated when the current solution S" is worse than the best solution found so far S (i.e. after Line 12 in Algorithm 2). However, the cooling scheme can be placed at the end of each iteration or at the end of each loop in the local search (the temperature is reset at the start of each local search). The former is the SAO whereas the latter is the SAI. A hybrid method is also proposed using the SAI

a)

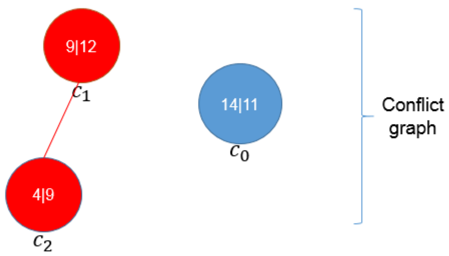
	Teachers																								
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	14	2	10	4	13	2	3	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	4	7	5	3	6	8	14	0	14	7	4	3	9	1	0	6



	Teachers																								
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	3	~~2~~	~~10~~	~~4~~	~~13~~	~~2~~	14	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	~~7~~	~~5~~	~~3~~	~~6~~	~~8~~	4	0	14	7	4	3	9	1	0	6

b)

	Teachers																								
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	3	2	10	4	13	2	14	4	3	1	0	14	11	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	7	5	3	6	8	4	0	14	7	4	3	9	1	0	6



	Teachers																								
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	9	4	1	0	6	11	13	9	3	2	10	4	13	2	11	~~4~~	~~3~~	~~1~~	~~0~~	~~14~~	14	6	10	8
c_1	14	5	7	9	4	7	0	13	5	12	13	14	8	8	7	9	6	3	4	0	3	12	4	7	6
c_2	9	13	13	7	8	1	7	5	4	14	7	5	3	6	8	4	0	14	7	4	3	9	1	0	6

Figure 3.7: S3R applied to one schedule with teachers' ids. a) TQ move applied to the current schedule. b) Another TQ move applied to the new schedule with timeslot t_{15} as the common timeslot. Conflicts are highlighted in blue and red in the conflict graphs.

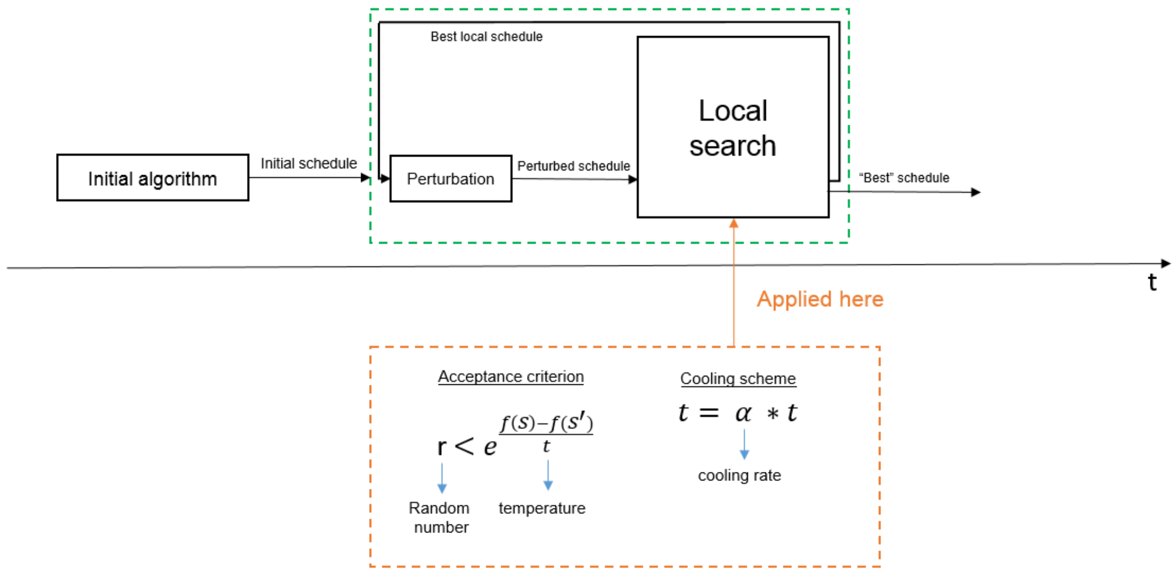


Figure 3.8: SAI overview. Both the acceptance criterion and the cooling scheme are applied inside the local search, the latter is applied at the end of the local search.

cooling scheme and the 2TQ perturbation operator.

Figure 3.8 and Figure 3.9 show the difference between the SAI and the SAO. The difference is that the former applies the cooling scheme inside the local search (i.e. after Line 15 in Algorithm 2), whereas the latter applies the cooling scheme outside and after the local search (i.e. after Line 18 in Algorithm 3).

3.7 HSTTP Instance Creator

Creating instances following the XML format used for this problem is a tedious task since there are multiples requirements per class and teacher unavailabilities. On this creator as seen in Figure 3.10, users can enter the number of classes, teachers, days, and periods for the schedule. After that, the "Generate" button generates two matrices of size $T \times 3$ and $T \times D \times P$ respectively where T is the number of teachers, D is the number of days, and P is the number of periods per day. In the first matrix, users can create requirements by choosing a class in the combobox, the teacher row, and entering the $Lessons$ (i.e. total number of lessons), Max (i.e. maximum number lessons per day), $DLessons$ (i.e. minimum number of double lessons). Requirements that have 0 lessons are ignored. In the second matrix, the user can define teachers' unavailabilities by clicking on any button that corresponds to the teacher, the day (i.e. the number in each button) and the period (the number above each button); if the button is green, it means that the teacher is available on that period, if it is red, it means that the teacher is unavailable on that period. Then, the user can click on the "Finish" button to generate the XML file. This form was created in C# on Visual Studio Community 2015.

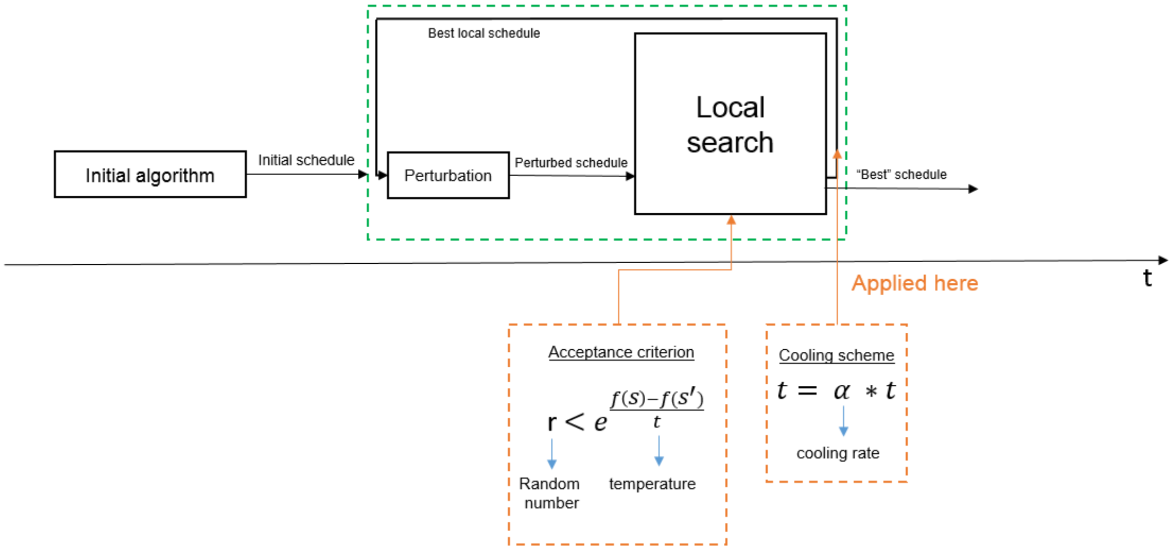


Figure 3.9: SAO overview. Only the acceptance criterion is inside the local search, but the cooling scheme is applied outside the local search at the end of the iteration of the meta-heuristic.

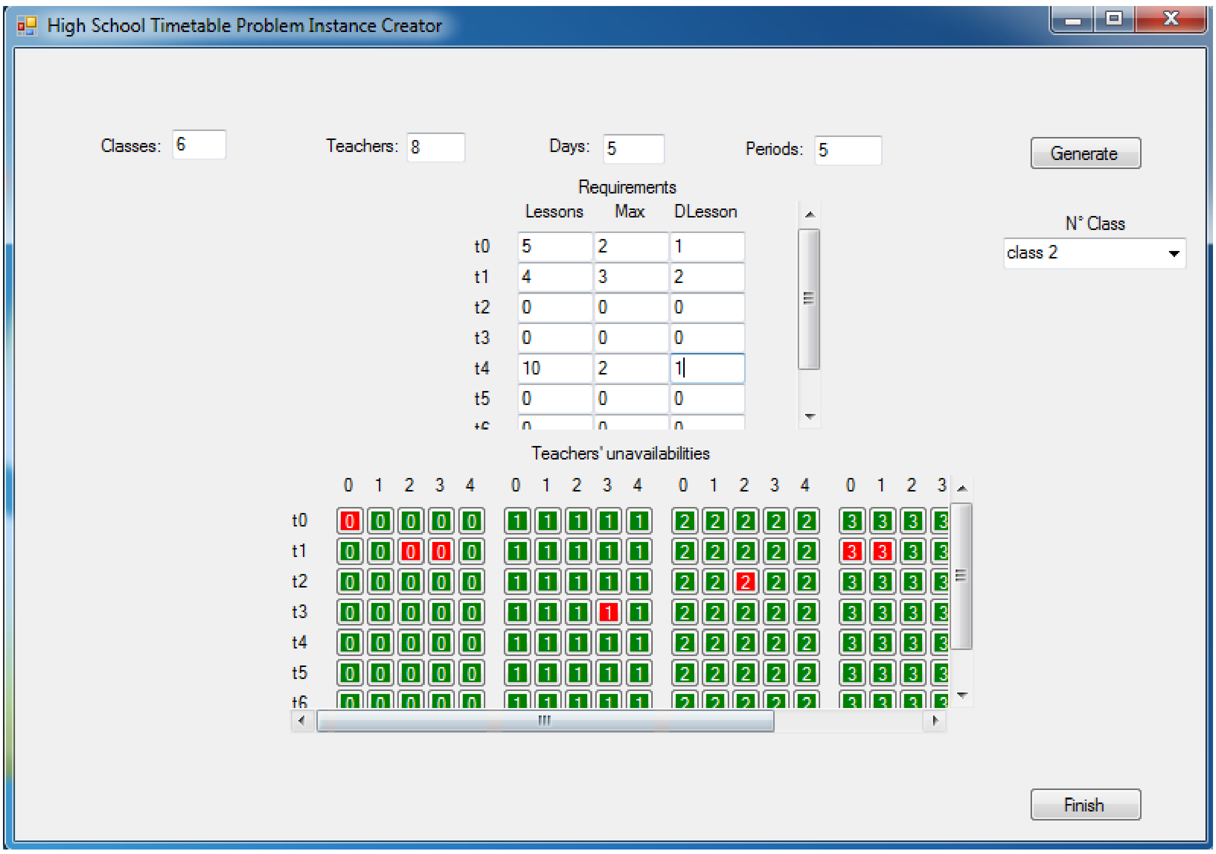


Figure 3.10: HSTTP instance creator user interface

```xml
<file>
    <data>
        <entities>
             <classes from="0" to="5"/>
             <teachers from="0" to="7"/>
             <days from="0" to="4"/>
             <periods from="0" to="4"/>
        </entities>

        <requirements>
            <requirement class="0" teacher="0" lessons="5" max_per_day="2" double_lessons="1"/>
            <requirement class="2" teacher="0" lessons="5" max_per_day="2" double_lessons="1"/>
            <requirement class="2" teacher="1" lessons="4" max_per_day="3" double_lessons="2"/>
            <requirement class="2" teacher="4" lessons="10" max_per_day="2" double_lessons="1"/>
        </requirements>

        <teacherunavailabilities>
            <unavailability teacher="0" day="0" period="0"/>
            <unavailability teacher="1" day="0" period="2"/>
            <unavailability teacher="1" day="0" period="3"/>
            <unavailability teacher="1" day="3" period="0"/>
            <unavailability teacher="1" day="3" period="1"/>
            <unavailability teacher="2" day="2" period="2"/>
            <unavailability teacher="3" day="1" period="3"/>
        </teacherunavailabilities>
    </data>
</file>
```


Figure 3.11: HSTTP instance creator XML output

Figure 3.11 shows the XML output of the creator given the input parameters shown in Figure 3.10, class 0 also has requirements that cannot be seen in the latter figure. Although this instance is not valid, it is still shown to give an idea about what this creator generates.

3.8 Final Considerations

The method proposed by Saviniec and Constantino (2017) does not mention how the constraint violations are calculated or what data structures they used to calculate them, so we mentioned what data structures we used. Our modifications are done to the perturbation phase, the original method uses one random TQ move, but the main problem with it is that it might get canceled in the local search. So, we are going to implement the following: two TQ moves, Relaxed rule, Random swaps per class, Change random column, and Swap Three Requirements. In the following section, the results to these changes are compared to the original method. Furthermore, the HSTTP instance creator's purpose is to help create instances in XML following this problem's format.

Chapter 4

Tests and Results

In this section, the performance of all methods is evaluated. The experimental computations, including running Saviniec et al.'s executable, were done in the High Computational Performance Center of the Peruvian Amazon from the the Investigation Institute of the Peruvian Amazon. The tests were run in nodes that had 28 cores (one test per core), Intel Xeon CPU, and 64 GB of RAM. The current implementations were coded in C++ and compiled with g++ 4.8.5 20150623 (Red Hat 4.8.5-4). In each experiment, 25 trials, for each instance, have been carried out. There is a total of 34 instances taken from Saviniec and Constantino (2017). Each instance has been run for 600 seconds which is 10 minutes.

4.1 Test Planning

The tests for each method consists in finding a near-optimum schedule for every instance after it has run for 10 minutes. The score of that schedule found is calculated according to the objection function. This is done 25 times for each instance and its average and percentage Average Relative Deviation (ARD) are calculated which has the following formula: $(\frac{Average}{Best} - 1) \times 100$ where $Average$ is the average value of all 25 values and $Best$ is the best value of all 25 values. This formula was obtained from: $\frac{100}{N} \sum_{i=1}^{n} \frac{|x_i - Best|}{Best}$ where x_i is the calculated value for the ith schedule and N is the total number of trials. This percentage ARD shows how much the average differs from the best value: the lower the percentage ARD, the closer the average is to the best solution. In case of the methods that have parameters, the same method is executed with different parameters for all instances and compared to the original method by Saviniec et al. The "Best" value is the best score obtained for that instance after running 25 trials. After that, every method is compared to each other including the one by Saviniec et al. A graph comparing both the best method and our ILS-TQ implementation for instances 33 and 34 is shown. At the end, we show how schedules evolve in some of these methods at some iterations.

4.2 Instance Descriptions

These 34 real-case instances are from years 2008, 2010, and 2011 which were collected from thirteen schools in Brazil. Many teachers in these instances work in more than one school which are located in different towns, so, getting a schedule that satisfies all teacher's unavailable periods was the most complicated (Saviniec & Constantino, 2017).

These instances are shown in Table 4.1 where $|C|$ means the number of classes, $|T|$ is the number of teachers, $|D|$ is the number of days, $|H|$ is the number of hours, $|U|$ is the number of unavailable periods, θ is the number of weekly meetings, and μ is the number of weekly double lessons.

4.3 Results of the Original Method

Table 4.2 shows all results obtained by running the original implementation by Saviniec and Constantino (2017) 25 times for each instance. This method obtained all feasible schedules for all instances; it got 25 best solutions for instances JNS-CEJXXIII-2011-N and CM-CEDB-2010-N and 24 best solutions for instance JNS-CEJXXIII-2011-M. The last instances only have one best solution found. The gray cells show the best value found for that instance.

Table 4.1: Features of the 34 instances

Instance	$\|C\|$	$\|T\|$	$\|D\|$	$\|H\|$	$\|U\|$	$\sum_{e\in R}\theta_e$	$\sum_{e\in R}\mu_e$	Size
CM-CEUP-2011-N	3	15	5	5	284	75	36	S
FA-EEF-2011-M	4	12	5	5	160	100	42	S
JNS-CEJXXIII-2011-N	4	15	5	5	12	100	48	S
CM-CEDB-2010-N	5	17	5	5	41	125	60	S
JNS-CEJXXIII-2011-M	5	18	5	5	50	125	60	S
JNS-CEJXXIII-2011-V	5	18	5	5	52	125	60	S
JNS-CEDPII-2011-V	7	21	5	5	101	175	73	S
CM-CECM-2011-N	8	30	5	5	489	200	96	S
JNS-CEDPII-2011-M	8	19	5	5	91	200	85	S
CL-CECL-2011-N-A	9	28	5	5	25	225	107	S
MGA-CEVB-2011-V	9	20	5	5	214	225	97	S
MGA-CEVB-2011-M	10	21	5	5	167	250	108	S
CL-CEASD-2008-V-A	12	27	5	5	108	300	132	M
CL-CEASD-2008-V-B	12	27	5	5	108	300	132	M
MGA-CEDC-2011-V	12	31	5	5	412	300	131	M
CL-CECL-2011-M-A	13	31	5	5	23	325	144	M
CL-CECL-2011-M-B	13	31	5	5	8	325	143	M
CM-CECM-2011-V	13	34	5	5	455	325	142	M
CL-CECL-2011-V-A	14	29	5	5	21	350	164	M
CM-CEUP-2008-V	16	35	5	5	345	400	192	M
CM-CEUP-2011-M	16	38	5	5	498	400	192	M
CM-CEUP-2011-V	16	34	5	5	382	400	169	M
MGA-CEJXXIII-2010-V	16	35	5	5	309	400	192	M
NE-CESVP-2011-V-A	16	44	5	5	181	400	183	M
NE-CESVP-2011-V-B	16	43	5	5	192	400	184	M
NE-CESVP-2011-V-C	16	43	5	5	218	400	182	M
NE-CESVP-2011-M-A	18	45	5	5	156	450	212	M
NE-CESVP-2011-M-B	18	44	5	5	167	450	212	M
NE-CESVP-2011-M-C	18	45	5	5	152	450	211	M
NE-CESVP-2011-M-D	18	45	5	5	267	450	211	M
MGA-CEDC-2011-M	19	37	5	5	382	475	210	M
CM-CECM-2011-M	20	51	5	5	648	500	234	M
MGA-CEGV-2011-M	31	62	5	5	588	775	352	L
MGA-CEGV-2011-V	32	75	5	5	857	800	357	L

Table 4.2: Results for the 25 runs of Saviniec et al.'s implementation of the ILS-TQ. Gray cells represent the best solutions.

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	**269**	**269**	**269**	**269**	270	270	**269**	270	**269**	270	**269**	**269**	**269**	270	270	270	270	270	**269**	270	**269**	**269**	**269**	270	270
FA-EEF-2011-M	255	255	255	255	255	255	255	255	256	255	255	255	255	255	255	255	255	255	255	255	255	255	255	**254**	**254**
JNS-CEJXXIII-2011-N	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**	**254**
CM-CEDB-2010-N	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**	**298**
JNS-CEJXXIII-2011-M	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	320	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**	**319**
JNS-CEJXXIII-2011-V	**323**	**323**	**323**	324	**323**	**323**	**323**	324	**323**	**323**	**323**	**323**	326	**323**	**323**	**323**	**323**	**323**	**323**	**323**	**323**	**323**	324	324	**323**
JNS-CEDPII-2011-V	460	459	**457**	461	459	461	458	461	458	463	460	464	458	463	460	462	462	462	460	460	461	461	458	464	460
CM-CECM-2011-N	674	673	683	679	689	682	679	684	671	676	679	672	691	677	686	**667**	677	674	676	685	677	674	673	675	677
JNS-CEDPII-2011-M	485	486	485	487	486	490	486	484	485	**481**	**481**	487	495	483	**481**	483	484	490	482	488	485	**481**	488	**481**	482
CL-CECL-2011-N-A	**627**	629	**627**	**627**	**627**	**627**	**627**	**627**	**627**	629	**627**	**627**	**627**	629	**627**	**627**	629	**627**	**627**	**627**	**627**	629	**627**	**627**	**627**
MGA-CEVB-2011-V	558	555	555	559	555	554	555	**552**	555	555	557	**552**	556	556	555	553	555	555	**552**	553	556	**552**	553	558	559
MGA-CEVB-2011-M	574	588	575	576	580	584	600	**571**	576	577	596	577	572	575	**571**	575	572	587	574	577	589	**571**	573	586	574
CL-CEASD-2008-V-A	704	705	702	702	704	**699**	701	702	**699**	704	703	704	705	709	701	707	700	701	707	703	700	703	701	704	705
CL-CEASD-2008-V-B	706	707	707	703	703	**699**	702	701	706	706	710	701	702	701	702	706	703	704	704	713	705	704	708	713	703
MGA-CEDC-2011-V	730	729	735	729	730	733	732	**728**	731	738	731	731	732	732	733	732	733	731	732	730	734	729	729	730	731
CL-CECL-2011-M-A	745	**744**	746	747	746	745	746	747	749	745	747	746	745	747	745	748	745	745	750	747	747	**744**	747	746	748
CL-CECL-2011-M-B	739	739	**736**	740	743	742	738	740	738	740	737	741	741	739	741	740	**736**	739	740	737	738	738	740	740	739
CM-CECM-2011-V	814	823	813	817	**806**	822	820	824	817	820	816	811	823	808	832	812	817	823	820	813	818	822	813	820	826
CL-CECL-2011-V-A	775	773	774	773	773	773	777	775	776	774	776	**772**	773	774	773	773	776	774	773	774	**772**	776	784	**772**	774
CM-CEUP-2008-V	**985**	988	1002	996	989	1001	991	986	993	991	989	995	994	988	989	1003	988	**985**	988	992	989	990	993	986	1003
CM-CEUP-2011-M	1039	1039	**1028**	1037	1033	1034	1040	1035	1034	1035	1045	1033	1039	1030	1031	1038	1043	1035	1039	1038	1035	1044	1035	1046	1029
CM-CEUP-2011-V	957	955	948	959	941	**940**	950	950	948	955	**940**	949	953	950	950	954	946	949	948	942	954	942	944	950	957
MGA-CEJXXIII-2010-V	941	937	940	942	936	931	926	951	946	944	924	936	940	937	**923**	938	927	940	929	930	933	934	933	942	935
NE-CESVP-2011-V-A	1032	1030	1033	**1027**	1037	1042	1030	1035	**1027**	1040	1031	1033	1037	1032	1035	1033	1037	1036	1032	**1027**	1037	1032	1030	1030	1038
NE-CESVP-2011-V-B	1036	1035	**1028**	1045	1032	1034	1036	1040	1036	1042	1038	1034	1036	1033	1041	1036	1036	1050	1039	1042	1034	1039	1031	1032	1036
NE-CESVP-2011-V-C	1026	1031	1030	1025	1023	1021	1028	1035	1033	1028	1032	1028	1030	1034	1028	1028	1037	1030	1030	1029	1022	1033	1026	1025	**1020**
NE-CESVP-2011-M-A	1131	1134	1135	1142	1143	1151	1136	1139	1137	1131	1138	1140	1136	1135	1138	1136	1137	1137	1136	1137	**1130**	1142	1143	1131	1141
NE-CESVP-2011-M-B	1134	1137	1131	1130	**1121**	1133	1143	1129	1134	1128	1125	1127	1129	1129	1131	1125	1133	1123	1134	1139	1130	1140	1132	1126	1130
NE-CESVP-2011-M-C	1142	1152	1148	1150	1144	1149	1148	1145	1152	1152	1154	1149	1147	1155	1147	1159	1152	1164	1150	1146	1158	**1141**	1158	1143	1147
NE-CESVP-2011-M-D	1140	1146	1145	1150	1161	1153	1148	1150	**1138**	1144	1146	1143	1145	1152	1146	1145	1145	1143	1150	1144	1144	1148	1148	1149	1147
MGA-CEDC-2011-M	1074	1074	1065	1069	1060	**1058**	1072	1069	1060	1065	1069	1065	1065	1069	1069	1061	1060	1066	1065	1063	1071	**1058**	1065	1063	1074
CM-CECM-2011-M	1255	1246	1251	**1243**	1261	1257	1256	1255	1253	1265	1257	1255	1257	1245	1247	1273	1248	1251	1250	1253	1261	1262	1257	1250	1256
MGA-CEGV-2011-M	1891	1869	1892	1880	1892	1884	1890	1870	1869	1883	1882	1874	1886	1880	1898	1876	1867	1884	1878	**1865**	1879	1867	1871	1886	1899
MGA-CEGV-2011-V	2082	2059	**2037**	2067	2038	2061	2061	2054	2099	2056	2041	2086	2054	2059	2042	2077	2082	2075	2069	2053	2044	2054	2051	2051	2050

31

4.4 Results of our Implementation of the Original ILS-TQ

Table 4.3 shows all results obtained by running our implementation of the original ILS-TQ by Saviniec and Constantino (2017) 25 times for each instance. This method obtained at least an unfeasible solution for instances CM-CEUP-2011-N, MGA-CEDC-2011-V, CM-CEUP-2011-M, CM-CEUP-2011-V, CM-CECM-2011-M and MGA-CEGV-2011-V. However, there were at most two hard constraints that were not satisfied in all these schedules. Furthermore, there were 28 instances in which every trial resulted in a feasible schedule (e.g. FA-EEF-2011-M), the instance JNS-CEJXXIII-2011-N got 25 best schedules and is the instance with the most number of best schedules. The gray cells show the best value found for that instance whereas the red cells show values for unfeasible solutions.

Table 4.3: Results for the 25 runs of our implementation of the ILS-TQ. Gray cells represent the best solutions; red cells represent unfeasible solutions.

	Runs																								
Instance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	269	269	269	269	269	270	269	270	269	269	269	270	269	269	269	269	269	270	269	269	270	269	269	270	269
FA-EEF-2011-M	256	255	255	256	255	255	257	255	257	255	255	255	255	255	255	256	255	255	254	255	255	255	255	255	255
JNS-CEJXXIII-2011-N	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	299	299	298	298	299	298	298	298	298	298	298	299	298	298	298	298	298	298	298	298	298	298	298	298
JNS-CEJXXIII-2011-M	323	321	320	319	320	323	320	324	322	319	319	321	320	319	319	319	321	319	319	323	319	319	320	319	319
JNS-CEJXXIII-2011-V	323	325	327	323	323	324	323	327	323	324	325	323	325	323	324	324	323	323	328	323	323	324	323	323	323
JNS-CEDPII-2011-V	461	469	461	463	462	464	464	466	462	463	463	464	463	480	464	461	462	465	465	464	468	464	464	471	461
CM-CECM-2011-N	680	675	687	674	679	694	692	685	695	680	676	683	684	669	693	676	677	694	677	677	684	672	100674	686	684
JNS-CEDPII-2011-M	484	500	488	486	488	495	485	503	489	482	492	495	483	492	486	489	490	481	490	486	488	500	492	505	488
CL-CECL-2011-N-A	631	633	635	629	636	630	630	628	631	631	634	631	628	631	634	631	628	632	631	636	627	632	631	636	630
MGA-CEVB-2011-V	559	560	559	564	560	563	565	565	557	562	561	556	561	556	562	558	558	562	558	554	560	559	561	563	557
MGA-CEVB-2011-M	595	593	579	598	579	586	590	588	586	580	580	585	581	605	584	587	589	581	576	587	588	598	588	583	589
CL-CEASD-2008-V-A	712	712	712	716	712	712	704	721	713	714	718	714	710	713	712	708	717	711	714	708	712	714	713	723	716
CL-CEASD-2008-V-B	714	708	710	721	715	723	706	713	711	711	713	711	716	715	718	708	709	711	708	721	715	715	702	714	700
MGA-CEDC-2011-V	743	734	748	100741	738	739	100733	740	745	734	735	100745	739	735	734	735	738	738	738	200749	739	744	743	742	737
CL-CECL-2011-M-A	754	750	759	756	762	758	752	759	759	749	757	757	756	754	758	756	759	752	764	753	763	755	757	754	764
CL-CECL-2011-M-B	742	748	755	747	754	752	750	761	752	760	745	746	748	746	751	749	756	749	756	751	750	748	752	748	758
CM-CECM-2011-V	830	822	829	840	830	829	827	827	833	829	840	832	825	826	829	831	818	829	830	833	827	828	833	831	837
CL-CECL-2011-V-A	795	789	794	789	786	798	781	793	797	793	785	783	794	784	801	793	791	783	797	795	809	794	784	789	793
CM-CEUP-2008-V	1026	1017	1019	1019	1019	1012	1005	1026	1022	1006	1012	1008	1039	1017	1022	1016	1024	1012	1020	1000	1033	1012	1031	1004	1017
CM-CEUP-2011-M	1051	1048	1052	1054	1063	1047	1044	1052	101050	1051	1048	1052	1047	1048	1050	1054	1057	1051	1052	1065	1067	1056	1066	1039	1055
CM-CEUP-2011-V	961	962	962	952	976	962	954	970	962	957	968	100951	963	944	959	965	963	959	100958	964	968	965	963	959	962
MGA-CEJXXIII-2010-V	957	944	967	962	945	954	952	941	965	942	947	956	944	957	956	979	964	942	969	956	956	944	961	944	964
NE-CESVP-2011-V-A	1059	1056	1062	1057	1046	1060	1053	1037	1066	1056	1057	1057	1054	1056	1041	1059	1057	1047	1058	1048	1064	1056	1057	1035	1050
NE-CESVP-2011-V-B	1054	1046	1056	1056	1046	1063	1058	1058	1046	1061	1052	1047	1045	1054	1050	1056	1065	1062	1059	1057	1058	1061	1065	1055	1055
NE-CESVP-2011-V-C	1047	1057	1043	1050	1059	1043	1040	1054	1049	1047	1067	1041	1050	1043	1038	1039	1040	1049	1041	1033	1062	1052	1041	1054	1051
NE-CESVP-2011-M-A	1154	1160	1170	1154	1160	1163	1175	1160	1168	1159	1164	1161	1160	1170	1173	1170	1175	1160	1159	1158	1173	1153	1178	1171	1161
NE-CESVP-2011-M-B	1175	1159	1156	1158	1151	1155	1167	1164	1152	1160	1156	1159	1162	1164	1175	1157	1182	1155	1158	1164	1167	1168	1162	1153	1167
NE-CESVP-2011-M-C	1163	1184	1202	1157	1180	1186	1174	1181	1178	1178	1177	1187	1174	1171	1187	1161	1175	1186	1182	1182	1182	1170	1173	1164	1184
NE-CESVP-2011-M-D	1172	1191	1188	1185	1171	1188	1185	1180	1183	1177	1168	1179	1171	1182	1179	1187	1184	1174	1181	1184	1185	1163	1166	1176	1179
MGA-CEDC-2011-M	1093	1091	1091	1088	1080	1096	1091	1079	1088	1083	1085	1094	1083	1079	1101	1089	1086	1088	1089	1081	1080	1083	1083	1092	1095
CM-CECM-2011-M	1285	1275	1274	1278	1279	1283	1300	1285	1293	1282	101273	1284	1286	1287	1280	1286	1281	1290	1295	1286	1314	1290	101279	1285	1280
MGA-CEGV-2011-M	1944	1949	1948	1934	1993	1978	1947	1953	1949	1956	1952	1938	1979	1940	1948	1955	1953	1972	1972	1973	1963	1951	1935	1930	1963
MGA-CEGV-2011-V	2139	2163	2156	2128	2127	2124	2158	2156	2145	102154	2142	2164	2156	2162	2142	2160	2145	2168	2154	2144	2143	2137	2110	2142	2151

33

Table 4.4 shows the best results, average and percentage ARD obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method, it shows how consistent the schedules produced by this method are, and if it produced at least one unfeasible schedule. For example, the best schedule for the instance CM-CECM-2011-N has a value of 669, but its average is 4681.88. The latter value was affected due to an unfeasible solution since it has a value greater than 10000. Moreover, the lower the average is, the more feasible solutions there are for that instance. For example, the instance MGA-CEDC-2011-V has four unfeasible solutions and has an average of 20739.4; unlike the last instance MGA-CEGV-2011-V that has one unfeasible solution and has an average of 6146.8. The maximum value for all instances that produced feasible schedules is 1.92931% for the instance JNS-CEDPII-2011-M, and for all instances including the ones that produced at least one unfeasible solution is 2725.54% for the instance MGA-CEDC-2011-V. Instance JNS-CEJXXIII-2011-N got a percentage ARD of 0.

4.5 Results of the Two ILS-TQ Moves (2TQ) Method

Table 4.5 shows all results obtained by running the 2TQ 25 times for each instance. It had problems dealing with hard constraints in instances MGA-CEDC-2011-V (19 feasible solutions), CM-CEUP-2011-M (22 feasible solutions) and CM-CECM-2011-M (24 feasible solutions). However, there was at most one hard constraint that was not satisfied in all these schedules. Furthermore, there were 31 instances in which every trial resulted in a feasible schedule (e.g. CM-CEUP-2011-N), the instance JNS-CEJXXIII-2011-N got 25 best schedules and is the instance with the most number of best schedules. The gray cells show the best value found for that instance whereas red cells show values for unfeasible solutions.

Table 4.4: Best results, average and percentage ARD, with its score for each constraint, out of the 25 runs of our implementation of the ILS-TQ

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	269.24	0.0892193
FA-EEF-2011-M	0	0	0	5	15	234	254	255.24	0.488189
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254	0
CM-CEDB-2010-N	0	0	0	10	0	288	298	298.16	0.0536913
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	320.24	0.388715
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	323.96	0.297214
JNS-CEDPII-2011-V	0	0	0	8	3	450	461	464.56	0.772234
CM-CECM-2011-N	0	0	0	24	33	612	669	4681.88	599.833
JNS-CEDPII-2011-M	0	0	0	10	3	468	481	490.28	1.92931
CL-CECL-2011-N-A	0	0	0	18	6	603	627	631.44	0.708134
MGA-CEVB-2011-V	0	0	0	14	0	540	554	560	1.08303
MGA-CEVB-2011-M	0	0	0	18	0	558	576	587	1.90972
CL-CEASD-2008-V-A	0	0	0	23	15	666	704	713.24	1.3125
CL-CEASD-2008-V-B	0	0	0	19	15	666	700	712.32	1.76
MGA-CEDC-2011-V	0	0	0	29	21	684	734	20739.4	2725.54
CL-CECL-2011-M-A	0	0	0	20	18	711	749	756.68	1.02537
CL-CECL-2011-M-B	0	0	0	25	6	711	742	750.96	1.20755
CM-CECM-2011-V	0	0	0	32	21	765	818	829.8	1.44254
CL-CECL-2011-V-A	0	0	0	31	3	747	781	791.6	1.35723
CM-CEUP-2008-V	0	0	0	61	21	918	1000	1017.52	1.752
CM-CEUP-2011-M	0	0	0	49	45	945	1039	5052.76	386.31
CM-CEUP-2011-V	0	0	0	32	21	891	944	8961.16	849.275
MGA-CEJXXIII-2010-V	0	0	0	50	9	882	941	954.72	1.45802
NE-CESVP-2011-V-A	0	0	0	36	27	972	1035	1053.92	1.82802
NE-CESVP-2011-V-B	0	0	0	37	36	972	1045	1055.4	0.995215
NE-CESVP-2011-V-C	0	0	0	34	36	963	1033	1047.6	1.41336
NE-CESVP-2011-M-A	0	0	0	49	33	1071	1153	1164.36	0.985256
NE-CESVP-2011-M-B	0	0	0	53	18	1080	1151	1161.84	0.94179
NE-CESVP-2011-M-C	0	0	0	47	21	1089	1157	1177.52	1.77355
NE-CESVP-2011-M-D	0	0	0	53	21	1089	1163	1179.12	1.38607
MGA-CEDC-2011-M	0	0	0	50	21	1008	1079	1087.52	0.78962
CM-CECM-2011-M	0	0	0	68	27	1179	1274	9285.2	628.823
MGA-CEGV-2011-M	0	0	0	109	39	1782	1930	1955	1.29534
MGA-CEGV-2011-V	0	0	0	106	51	1953	2110	6146.8	191.318

Table 4.5: Results for the 25 runs of the method 2TQ. Gray cells represent the best solutions; red cells represent unfeasible solutions.

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	269	269	269	269	269	269	269	270	269	269	269	269	269	269	269	269	269	269	269	269	269	269	269	269	269
FA-EEF-2011-M	256	255	255	255	255	255	255	255	255	256	255	255	255	255	255	255	254	256	257	255	255	255	255	255	255
JNS-CEJXXIII-2011-N	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	298	298	298	298	298	298	298	298	298	298	298	298	299	298	298	298	298	298	298	298	298	298	298	302
JNS-CEJXXIII-2011-M	319	319	323	320	320	319	322	319	320	323	319	319	320	322	319	320	319	321	319	320	319	322	319	319	319
JNS-CEJXXIII-2011-V	324	323	323	323	325	324	324	323	326	325	323	323	323	323	324	323	323	323	323	323	324	323	324	325	323
JNS-CEDPII-2011-V	462	464	463	471	462	462	463	470	465	461	462	469	463	467	464	465	461	463	464	464	468	462	460	462	466
CM-CECM-2011-N	682	681	681	676	676	687	682	679	691	680	674	686	678	690	687	678	672	686	684	678	681	682	679	679	674
JNS-CEDPII-2011-M	482	488	491	485	489	486	487	484	483	494	488	483	486	490	496	505	487	490	483	490	484	484	493	490	489
CL-CECL-2011-N-A	630	634	631	631	628	636	631	629	631	632	633	628	634	630	640	632	630	632	632	630	634	630	630	631	630
MGA-CEVB-2011-V	560	561	562	560	562	559	559	561	562	555	563	562	556	562	560	557	560	563	560	559	559	560	558	560	558
MGA-CEVB-2011-M	579	595	586	590	590	601	592	595	578	583	589	588	585	584	588	591	598	586	578	577	592	593	600	583	579
CL-CEASD-2008-V-A	716	720	716	714	712	717	716	719	723	717	716	715	716	717	714	717	708	717	711	710	712	714	720	712	709
CL-CEASD-2008-V-B	711	710	714	718	715	708	713	715	719	720	711	709	714	714	709	722	718	711	714	719	712	715	716	713	709
MGA-CEDC-2011-V	740	738	736	743	100736	100741	742	738	100738	100737	742	734	736	100730	738	737	745	736	741	738	732	738	100744	738	740
CL-CECL-2011-M-A	759	758	758	754	756	758	765	760	764	764	764	758	760	761	762	762	757	763	756	756	759	758	756	764	760
CL-CECL-2011-M-B	754	746	759	752	746	751	748	756	749	755	745	751	754	747	760	752	752	751	750	745	758	748	751	750	747
CM-CECM-2011-V	826	811	829	823	836	835	830	822	835	827	821	831	833	822	825	819	829	834	836	830	829	832	833	843	834
CL-CECL-2011-V-A	783	796	793	785	795	796	788	787	793	788	783	784	785	788	800	801	803	790	797	798	786	806	787	788	807
CM-CEUP-2008-V	1006	1021	1020	1027	1020	1032	1010	1007	1014	1022	1013	1017	1002	1019	1008	1027	1026	1028	1011	1010	1021	1008	1005	1021	1011
CM-CEUP-2011-M	1060	1052	1048	1057	1059	101053	1064	1053	1050	1056	101047	1049	1052	1062	1062	1064	1050	1051	1054	1050	1059	1055	101052	1055	1046
CM-CEUP-2011-V	972	973	961	971	967	966	968	964	953	965	966	965	969	959	959	970	956	955	967	962	969	954	968	967	957
MGA-CEJXXIII-2010-V	962	963	948	949	973	953	957	944	935	938	966	960	952	950	950	958	942	964	976	966	952	955	956	966	957
NE-CESVP-2011-V-A	1058	1048	1062	1046	1060	1041	1067	1054	1057	1052	1051	1050	1056	1063	1056	1059	1068	1053	1063	1046	1057	1048	1047	1078	1059
NE-CESVP-2011-V-B	1044	1057	1054	1058	1071	1053	1058	1052	1063	1041	1053	1056	1054	1047	1070	1078	1060	1060	1059	1074	1065	1050	1061	1049	1058
NE-CESVP-2011-V-C	1046	1065	1054	1048	1043	1047	1058	1046	1047	1057	1053	1052	1053	1038	1060	1045	1057	1064	1055	1064	1043	1037	1063	1035	1045
NE-CESVP-2011-M-A	1158	1169	1168	1176	1167	1172	1166	1161	1170	1167	1177	1187	1164	1170	1165	1170	1169	1175	1162	1161	1172	1159	1168	1168	1162
NE-CESVP-2011-M-B	1163	1168	1148	1159	1168	1162	1155	1170	1158	1148	1156	1152	1166	1168	1166	1175	1166	1164	1152	1159	1155	1172	1155	1158	1148
NE-CESVP-2011-M-C	1182	1173	1197	1190	1181	1182	1185	1172	1185	1183	1167	1198	1171	1174	1169	1170	1184	1173	1181	1169	1176	1194	1181	1191	1184
NE-CESVP-2011-M-D	1181	1179	1174	1182	1173	1175	1171	1170	1188	1161	1181	1167	1188	1176	1168	1174	1179	1192	1161	1180	1181	1187	1178	1179	1172
MGA-CEDC-2011-M	1094	1105	1092	1096	1097	1086	1083	1091	1105	1105	1080	1084	1093	1087	1084	1092	1089	1081	1085	1086	1095	1094	1087	1089	1090
CM-CECM-2011-M	1277	1291	1287	1292	1286	1282	1302	1291	1287	1294	1301	11278	1268	1299	1292	1285	1261	1275	1273	1284	1271	1290	1300	1296	1280
MGA-CEGV-2011-M	1959	1961	1942	1982	1965	1961	1970	1977	1953	1970	1947	1947	1941	1952	1978	1985	1940	1946	1969	1944	1991	1954	1966	1926	1969
MGA-CEGV-2011-V	2147	2141	2129	2142	2156	2138	2153	2146	2139	2176	2148	2119	2143	2144	2150	2128	2149	2129	2176	2147	2159	2151	2165	2139	2145

Table 4.6 shows the best results, average and percentage ARD obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances that produced feasible schedules is 2.21948% for the instance CM-CECM-2011-V, and for all instances including the ones that produced at least one unfeasible solution is 3279.55% for the instance MGA-CEDC-2011-V. Instance JNS-CEJXXIII-2011-N got a percentage ARD of 0.

4.6 Results of the Relaxed Rule (RR)

Table 4.7 shows all results obtained by running the method RR 25 times for each instance. It found no unfeasible solutions. However, there was no instance that had all 25 best solutions and this method did not get the lower bound for any instance. The gray cells show the best value found for that instance and the red cells show values for unfeasible solutions.

Table 4.6: Best results, average and percentage ARD, with its score for each constraint, out of the 25 runs of the method 2TQ

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	269.04	0.0148699
FA-EEF-2011-M	0	0	0	5	15	234	254	255.16	0.456693
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254	0
CM-CEDB-2010-N	0	0	0	10	0	288	298	298.2	0.0671141
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	320	0.31348
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	323.6	0.185759
JNS-CEDPII-2011-V	0	0	0	10	0	450	460	464.12	0.895652
CM-CECM-2011-N	0	0	0	27	33	612	672	680.92	1.32738
JNS-CEDPII-2011-M	0	0	0	11	3	468	482	488.28	1.3029
CL-CECL-2011-N-A	0	0	0	19	6	603	628	631.56	0.566879
MGA-CEVB-2011-V	0	0	0	15	0	540	555	559.92	0.886486
MGA-CEVB-2011-M	0	0	0	16	3	558	577	588	1.90641
CL-CEASD-2008-V-A	0	0	0	24	18	666	708	715.12	1.00565
CL-CEASD-2008-V-B	0	0	0	24	18	666	708	713.96	0.841808
MGA-CEDC-2011-V	0	0	0	24	15	693	732	24738.3	3279.55
CL-CECL-2011-M-A	0	0	0	25	18	711	754	759.68	0.753316
CL-CECL-2011-M-B	0	0	0	25	9	711	745	751.08	0.816107
CM-CECM-2011-V	0	0	0	28	18	765	811	829	2.21948
CL-CECL-2011-V-A	0	0	0	36	0	747	783	792.28	1.18519
CM-CEUP-2008-V	0	0	0	57	18	927	1002	1016.24	1.42116
CM-CEUP-2011-M	0	0	0	56	45	945	1046	13054.4	1148.03
CM-CEUP-2011-V	0	0	0	35	27	891	953	964.12	1.16684
MGA-CEJXXIII-2010-V	0	0	0	44	18	873	935	955.68	2.21176
NE-CESVP-2011-V-A	0	0	0	42	27	972	1041	1055.96	1.43708
NE-CESVP-2011-V-B	0	0	0	42	27	972	1041	1057.8	1.61383
NE-CESVP-2011-V-C	0	0	0	42	39	954	1035	1051	1.54589
NE-CESVP-2011-M-A	0	0	0	51	36	1071	1158	1168.12	0.873921
NE-CESVP-2011-M-B	0	0	0	47	21	1080	1148	1160.44	1.08362
NE-CESVP-2011-M-C	0	0	0	54	24	1089	1167	1180.48	1.1551
NE-CESVP-2011-M-D	0	0	0	48	15	1098	1161	1176.68	1.35056
MGA-CEDC-2011-M	0	0	0	42	21	1017	1080	1090.8	1
CM-CECM-2011-M	0	0	0	64	36	1161	1261	1685.68	33.678
MGA-CEGV-2011-M	0	0	0	111	42	1773	1926	1959.8	1.75493
MGA-CEGV-2011-V	0	0	0	100	39	1980	2119	2146.36	1.29118

Table 4.7: Results for the 25 runs of the method RR. Gray cells represent the best solutions; red cells represent unfeasible solutions.

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	270	270	270	270	270	270	270	275	270	270	271	270	273	270	270	270	271	275	274	270	270	270	270	270	270
FA-EEF-2011-M	277	275	280	278	275	281	279	281	275	272	277	281	275	280	277	278	277	278	277	280	275	275	267	281	276
JNS-CEJXXIII-2011-N	277	275	273	271	278	269	272	266	275	271	279	279	271	274	275	276	274	271	273	275	272	270	271	278	269
CM-CEDB-2010-N	341	339	340	349	348	345	345	337	343	338	335	341	345	336	335	328	341	343	343	340	341	342	343	339	343
JNS-CEJXXIII-2011-M	353	348	355	352	360	358	352	351	359	350	347	345	349	359	354	359	357	331	344	350	353	358	359	358	364
JNS-CEJXXIII-2011-V	361	362	357	356	350	350	365	367	355	349	355	362	353	356	346	344	353	351	347	341	360	359	362	355	342
JNS-CEDPII-2011-V	532	532	534	528	543	546	541	546	523	533	540	534	529	543	529	536	534	541	541	536	535	537	544	537	513
CM-CECM-2011-N	703	706	692	696	705	708	706	714	706	698	688	713	700	694	705	705	701	703	708	701	693	705	708	699	710
JNS-CEDPII-2011-M	572	573	581	578	581	571	570	566	588	560	584	579	574	565	577	568	552	562	575	553	581	562	574	570	562
CL-CECL-2011-N-A	765	759	752	742	753	757	745	764	744	756	752	752	755	753	751	763	753	757	759	763	767	751	753	753	756
MGA-CEVB-2011-V	663	665	665	666	666	660	659	659	664	664	662	661	662	669	664	666	664	667	665	665	667	666	666	663	665
MGA-CEVB-2011-M	692	691	697	702	710	689	701	707	702	701	688	705	710	706	705	703	703	708	693	715	703	689	709	708	692
CL-CEASD-2008-V-A	864	856	849	860	863	844	855	863	867	859	858	870	849	857	848	865	854	860	861	854	862	864	871	872	846
CL-CEASD-2008-V-B	859	837	849	861	860	859	864	874	868	856	846	865	869	850	871	866	871	851	860	862	868	871	855	868	855
MGA-CEDC-2011-V	821	830	822	819	831	823	824	830	828	833	821	828	827	832	818	828	824	830	832	833	826	832	826	836	816
CL-CECL-2011-M-A	1022	1013	1055	1046	1039	1044	1057	1042	1034	1046	1040	1042	1043	1039	1038	1026	1038	1041	1028	1037	1037	1042	1036	1032	1042
CL-CECL-2011-M-B	1040	1046	1045	1032	1047	1072	1061	1027	1055	1049	1053	1051	1054	1055	1056	1058	1037	1027	1046	1056	1044	1043	1050	1044	1041
CM-CECM-2011-V	931	915	932	919	919	921	918	911	930	915	914	920	918	913	924	936	931	935	937	911	918	922	923	912	931
CL-CECL-2011-V-A	1145	1123	1140	1125	1134	1116	1145	1136	1135	1134	1125	1124	1146	1144	1139	1137	1142	1124	1116	1133	1141	1090	1138	1117	1131
CM-CEUP-2008-V	1181	1163	1175	1171	1174	1181	1169	1179	1159	1179	1164	1170	1147	1174	1181	1160	1162	1146	1175	1178	1173	1140	1141	1124	1176
CM-CEUP-2011-M	1173	1167	1161	1162	1164	1157	1175	1164	1163	1168	1159	1168	1162	1169	1164	1160	1168	1160	1166	1148	1162	1169	1160	1160	1162
CM-CEUP-2011-V	1090	1089	1091	1094	1095	1093	1096	1084	1090	1093	1094	1094	1089	1090	1091	1089	1089	1095	1085	1092	1093	1089	1091	1092	1089
MGA-CEJXXIII-2010-V	1129	1124	1121	1120	1115	1115	1105	1102	1090	1120	1104	1115	1103	1107	1116	1107	1113	1126	1127	1114	1120	1121	1100	1108	1099
NE-CESVP-2011-V-A	1289	1294	1306	1287	1301	1308	1285	1312	1315	1302	1284	1302	1291	1295	1308	1308	1280	1306	1301	1301	1314	1306	1295	1294	1303
NE-CESVP-2011-V-B	1293	1298	1288	1296	1293	1290	1283	1292	1289	1288	1278	1284	1290	1285	1290	1288	1294	1280	1294	1297	1289	1271	1273	1281	1289
NE-CESVP-2011-V-C	1246	1241	1251	1264	1256	1258	1261	1250	1261	1248	1231	1263	1230	1276	1255	1286	1237	1239	1248	1244	1260	1270	1262	1262	1253
NE-CESVP-2011-M-A	1451	1472	1471	1485	1473	1473	1472	1469	1443	1459	1482	1461	1460	1477	1453	1465	1464	1465	1465	1446	1473	1486	1479	1454	1479
NE-CESVP-2011-M-B	1434	1442	1458	1443	1466	1458	1447	1453	1437	1448	1455	1448	1445	1441	1463	1468	1451	1453	1452	1449	1439	1449	1432	1456	1435
NE-CESVP-2011-M-C	1486	1506	1501	1495	1496	1493	1491	1506	1492	1498	1502	1502	1508	1490	1495	1501	1503	1469	1502	1505	1481	1519	1504	1514	1493
NE-CESVP-2011-M-D	1435	1418	1435	1437	1420	1436	1408	1413	1398	1425	1430	1428	1443	1427	1428	1433	1417	1403	1431	1434	1409	1428	1433	1437	1414
MGA-CEDC-2011-M	1258	1262	1253	1242	1251	1258	1246	1261	1257	1257	1250	1253	1264	1258	1254	1259	1257	1261	1247	1255	1253	1257	1258	1253	1250
CM-CECM-2011-M	1518	1521	1519	1523	1510	1510	1506	1499	1506	1510	1514	1518	1510	1518	1505	1503	1524	1497	1512	1507	1504	1502	1523	1505	1513
MGA-CEGV-2011-M	2419	2401	2396	2416	2413	2420	2422	2403	2395	2391	2421	2407	2408	2424	2432	2402	2417	2420	2424	2419	2421	2416	2431	2407	2403
MGA-CEGV-2011-V	2511	2487	2507	2510	2510	2509	2509	2505	2485	2498	2495	2518	2510	2511	2510	2507	2511	2504	2513	2502	2513	2523	2494	2504	2487

39

Table 4.8 shows the best results, average and percentage ARD obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances is 6.64653% for the instance JNS-CEJXXIII-2011-M.

4.7 Results of the Random Swaps per Class (SC) Method

Table 4.9 shows all results obtained by running the method SC 25 times for each instance. It had a lot more problems dealing with hard constraints in instance CM-CEUP-2011-M (seven unfeasible solutions). Surprisingly, it got all feasible solutions in the remaining instances. However, there was at most one hard constraint that was not satisfied in all these schedules. Furthermore, there were 33 instances in which every trial resulted in a feasible schedule (e.g. CM-CEUP-2011-N), the instance CM-CEUP-2011-N and JNS-CEJXXIII-2011-N got 24 best schedules. The gray cells show the best value found for that instance whereas red cells show values for unfeasible solutions.

Table 4.8: Best results, average and percentage ARD with its score for each constraint,out of the 25 runs of the method RR

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	0	0	0	270	270.76	0.281481
FA-EEF-2011-M	0	0	0	0	0	0	267	277.08	3.77528
JNS-CEJXXIII-2011-N	0	0	0	0	0	0	266	273.36	2.76692
CM-CEDB-2010-N	0	0	0	0	0	0	328	340.8	3.90244
JNS-CEJXXIII-2011-M	0	0	0	0	0	0	331	353	6.64653
JNS-CEJXXIII-2011-V	0	0	0	0	0	0	341	354.32	3.90616
JNS-CEDPII-2011-V	0	0	0	0	0	0	513	535.48	4.38207
CM-CECM-2011-N	0	0	0	0	0	0	688	702.68	2.13372
JNS-CEDPII-2011-M	0	0	0	0	0	0	552	571.12	3.46377
CL-CECL-2011-N-A	0	0	0	0	0	0	742	755	1.75202
MGA-CEVB-2011-V	0	0	0	0	0	0	659	664.12	0.776935
MGA-CEVB-2011-M	0	0	0	0	0	0	688	701.16	1.91279
CL-CEASD-2008-V-A	0	0	0	0	0	0	844	858.84	1.75829
CL-CEASD-2008-V-B	0	0	0	0	0	0	837	860.6	2.81959
MGA-CEDC-2011-V	0	0	0	0	0	0	816	826.8	1.32353
CL-CECL-2011-M-A	0	0	0	0	0	0	1013	1038.36	2.50346
CL-CECL-2011-M-B	0	0	0	0	0	0	1027	1047.56	2.00195
CM-CECM-2011-V	0	0	0	0	0	0	911	922.24	1.23381
CL-CECL-2011-V-A	0	0	0	0	0	0	1090	1131.2	3.77982
CM-CEUP-2008-V	0	0	0	0	0	0	1124	1165.68	3.70819
CM-CEUP-2011-M	0	0	0	0	0	0	1148	1163.64	1.36237
CM-CEUP-2011-V	0	0	0	0	0	0	1084	1091.08	0.653137
MGA-CEJXXIII-2010-V	0	0	0	0	0	0	1090	1112.84	2.09541
NE-CESVP-2011-V-A	0	0	0	0	0	0	1280	1299.48	1.52187
NE-CESVP-2011-V-B	0	0	0	0	0	0	1271	1287.72	1.3155
NE-CESVP-2011-V-C	0	0	0	0	0	0	1230	1254.08	1.95772
NE-CESVP-2011-M-A	0	0	0	0	0	0	1443	1467.08	1.66875
NE-CESVP-2011-M-B	0	0	0	0	0	0	1432	1448.88	1.17877
NE-CESVP-2011-M-C	0	0	0	0	0	0	1469	1498.08	1.97958
NE-CESVP-2011-M-D	0	0	0	0	0	0	1398	1424.8	1.91702
MGA-CEDC-2011-M	0	0	0	0	0	0	1242	1254.96	1.04348
CM-CECM-2011-M	0	0	0	0	0	0	1497	1511.08	0.940548
MGA-CEGV-2011-M	0	0	0	0	0	0	2391	2413.12	0.925136
MGA-CEGV-2011-V	0	0	0	0	0	0	2485	2505.32	0.817706

Table 4.9: Results for the 25 runs out of the method SC. Gray cells represent the best solutions; red cells represent unfeasible solutions.

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	269	269	269	269	269	269	269	269	269	269	269	270	269	269	269	269	269	269	269	269	269	269	269	269	269
FA-EEF-2011-M	255	254	255	255	255	255	255	255	256	256	255	255	255	255	255	255	255	255	255	255	255	255	256	255	255
JNS-CEJXXIII-2011-N	254	254	255	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	298	298	301	299	299	298	298	299	301	299	299	302	300	298	299	299	302	299	299	298	299	299	299	298
JNS-CEJXXIII-2011-M	322	322	325	323	325	319	319	323	319	323	324	322	323	321	326	322	324	320	320	319	322	323	321	323	324
JNS-CEJXXIII-2011-V	327	329	323	324	324	325	327	323	324	324	323	326	324	323	324	323	324	324	323	325	323	323	324	324	323
JNS-CEDPII-2011-V	473	470	468	469	477	472	472	471	471	469	473	469	472	470	466	474	470	468	469	473	469	470	472	471	467
CM-CECM-2011-N	688	685	690	668	682	681	699	711	688	699	680	696	684	694	699	687	688	699	696	690	685	679	688	683	691
JNS-CEDPII-2011-M	497	503	498	499	500	506	499	493	505	499	502	495	495	506	511	496	496	501	502	505	492	498	496	492	502
CL-CECL-2011-N-A	641	646	639	651	645	646	642	648	645	638	647	643	640	645	643	642	645	647	644	639	650	641	646	642	640
MGA-CEVB-2011-V	574	574	573	564	575	572	571	574	568	573	568	571	571	576	570	572	572	568	570	575	572	567	565	573	571
MGA-CEVB-2011-M	605	604	600	601	606	606	601	600	602	603	608	597	598	601	607	600	593	600	599	602	610	602	608	595	594
CL-CEASD-2008-V-A	741	731	736	745	729	736	742	742	733	738	738	735	741	740	730	734	740	748	732	737	736	741	730	739	739
CL-CEASD-2008-V-B	740	741	742	733	738	740	739	743	735	739	729	740	740	739	741	736	737	739	741	736	746	738	740	734	744
MGA-CEDC-2011-V	756	756	759	760	759	755	758	758	764	761	762	753	762	757	752	763	757	762	758	759	754	755	757	754	750
CL-CECL-2011-M-A	791	802	799	792	795	793	793	787	787	799	778	794	796	792	793	782	795	798	786	799	799	792	787	765	786
CL-CECL-2011-M-B	775	786	785	788	793	790	781	786	784	777	783	773	778	791	787	791	784	785	788	784	782	789	777	794	790
CM-CECM-2011-V	834	860	841	856	840	853	847	853	854	843	844	843	851	848	845	843	855	848	861	845	857	852	855	846	835
CL-CECL-2011-V-A	827	818	817	815	818	823	817	825	830	828	830	825	824	818	817	825	826	818	824	821	826	813	824	825	819
CM-CEUP-2008-V	1059	1054	1058	1056	1053	1061	1065	1069	1070	1045	1063	1053	1055	1067	1054	1061	1066	1048	1050	1056	1049	1063	1052	1058	1070
CM-CEUP-2011-M	1092	101091	1096	1098	1099	101093	1103	11087	1098	1088	1091	1098	101082	101091	1094	101087	1118	1089	1101	1114	101089	1091	1089	1084	1092
CM-CEUP-2011-V	992	986	982	989	995	987	992	996	990	993	991	989	984	991	987	985	990	988	994	995	999	980	992	988	997
MGA-CEJXXIII-2010-V	985	988	993	995	994	987	978	981	994	989	985	983	995	985	978	991	995	990	996	982	983	989	982	981	984
NE-CESVP-2011-V-A	1099	1090	1090	1100	1099	1099	1098	1096	1106	1106	1093	1100	1091	1087	1087	1093	1094	1097	1096	1102	1099	1101	1105	1094	1098
NE-CESVP-2011-V-B	1097	1101	1088	1088	1086	1093	1096	1097	1093	1093	1098	1087	1099	1087	1093	1102	1091	1094	1088	1107	1095	1089	1093	1073	1101
NE-CESVP-2011-V-C	1088	1085	1087	1095	1088	1084	1083	1093	1087	1090	1084	1086	1093	1094	1091	1082	1081	1101	1080	1094	1089	1082	1071	1093	1091
NE-CESVP-2011-M-A	1220	1217	1210	1212	1212	1216	1219	1217	1210	1221	1214	1209	1208	1209	1212	1219	1204	1225	1215	1211	1221	1208	1211	1220	1219
NE-CESVP-2011-M-B	1202	1218	1209	1216	1215	1212	1213	1212	1207	1204	1214	1216	1207	1207	1207	1195	1205	1205	1208	1203	1209	1206	1203	1205	1210
NE-CESVP-2011-M-C	1225	1219	1221	1235	1223	1236	1223	1216	1225	1225	1225	1224	1228	1225	1218	1223	1211	1222	1223	1231	1224	1222	1228	1226	1218
NE-CESVP-2011-M-D	1227	1207	1219	1215	1214	1231	1231	1212	1231	1211	1200	1223	1218	1222	1210	1221	1224	1217	1220	1222	1212	1225	1219	1215	1213
MGA-CEDC-2011-M	1124	1125	1124	1128	1115	1133	1127	1125	1122	1125	1115	1127	1129	1127	1125	1121	1125	1116	1129	1114	1128	1133	1120	1122	1125
CM-CECM-2011-M	1327	1330	1310	1335	1326	1328	1333	1332	1333	1324	1324	1327	1324	1341	1322	1337	1329	1329	1327	1329	1326	1336	1322	1332	1318
MGA-CEGV-2011-M	2017	2039	2021	2018	2031	2016	2025	2026	2012	2031	2004	2034	2041	2037	2024	2032	2026	2016	2012	2018	2031	2024	2022	2038	2024
MGA-CEGV-2011-V	2218	2218	2213	2221	2214	2215	2210	2205	2217	2199	2193	2215	2205	2210	2203	2207	2208	2211	2211	2184	2211	2209	2183	2202	2207

42

Table 4.10 shows the best results, average and percentage ARD obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances that produced feasible schedules is 3.42484% for the instance CL-CECL-2011-M-A, and for all instances including the ones that produced at least one unfeasible solution is 2251.86% for the instance CM-CEUP-2011-M.

4.8 Results of the Change Random Column (CC) Method

Table 4.11 shows all results obtained by running the method CC 25 times for each instance. This method only obtained one feasible schedule for the instance CM-CEUP-2011-N which is the smallest instance. It had a lot more problems dealing with hard constraints and it got zero feasible solutions in seven instances. Furthermore, there were also 14 instances in which every trial resulted in a feasible schedule (e.g. JNS-CEJXXIII-2011-N), unfortunately, all instances have at most one best schedule. This method obtained these results because it does not perturb the solution since the current solution and the best solution are always the same so the current solution is never modified and the local search gives the same result. In other words, it only locally searches once. However, these results show for what instances this method can find a feasible solution after only one local search. The gray cells show the best value found for that instance whereas red cells show values for unfeasible solutions.

Table 4.10: Best results, average and percentage ARD with its score for each constraint, out of the 25 runs of the method SC

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	269.04	0.0148699
FA-EEF-2011-M	0	0	0	5	15	234	254	255.08	0.425197
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254.04	0.015748
CM-CEDB-2010-N	0	0	0	10	0	288	298	299.12	0.375839
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	322.16	0.990596
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	324.24	0.383901
JNS-CEDPII-2011-V	0	0	0	10	6	450	466	470.6	0.987124
CM-CECM-2011-N	0	0	0	26	39	603	668	689.2	3.17365
JNS-CEDPII-2011-M	0	0	0	18	6	468	492	499.52	1.52846
CL-CECL-2011-N-A	0	0	0	26	9	603	638	643.8	0.909091
MGA-CEVB-2011-V	0	0	0	24	0	540	564	571.16	1.2695
MGA-CEVB-2011-M	0	0	0	26	9	558	593	601.68	1.46374
CL-CEASD-2008-V-A	0	0	0	36	27	666	729	737.32	1.14129
CL-CEASD-2008-V-B	0	0	0	39	24	666	729	738.8	1.34431
MGA-CEDC-2011-V	0	0	0	42	24	684	750	757.64	1.01867
CL-CECL-2011-M-A	0	0	0	36	18	711	765	791.2	3.42484
CL-CECL-2011-M-B	0	0	0	41	12	720	773	784.84	1.53169
CM-CECM-2011-V	0	0	0	36	33	765	834	848.36	1.72182
CL-CECL-2011-V-A	0	0	0	57	9	747	813	822.12	1.12177
CM-CEUP-2008-V	0	0	0	79	30	936	1045	1058.2	1.26316
CM-CEUP-2011-M	0	0	0	73	57	954	1084	25494.2	2251.86
CM-CEUP-2011-V	0	0	0	47	42	891	980	990.08	1.02857
MGA-CEJXXIII-2010-V	0	0	0	72	24	882	978	987.32	0.952965
NE-CESVP-2011-V-A	0	0	0	49	39	999	1087	1096.8	0.901564
NE-CESVP-2011-V-B	0	0	0	53	48	972	1073	1093.16	1.87884
NE-CESVP-2011-V-C	0	0	0	54	36	981	1071	1087.68	1.55742
NE-CESVP-2011-M-A	0	0	0	61	36	1107	1204	1214.36	0.860465
NE-CESVP-2011-M-B	0	0	0	61	54	1080	1195	1208.32	1.11464
NE-CESVP-2011-M-C	0	0	0	62	33	1116	1211	1223.84	1.06028
NE-CESVP-2011-M-D	0	0	0	66	27	1107	1200	1218.36	1.53
MGA-CEDC-2011-M	0	0	0	70	27	1017	1114	1124.16	0.912029
CM-CECM-2011-M	0	0	0	77	54	1179	1310	1328.04	1.3771
MGA-CEGV-2011-M	0	0	0	141	45	1818	2004	2024.76	1.03593
MGA-CEGV-2011-V	0	0	0	125	60	1998	2183	2207.56	1.12506

Table 4.11: Results for the 25 runs of the method CC. Gray cells represent the best solutions; red cells represent unfeasible solutions.

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	300280	100282	300268	500254	200279	100282	500286	200276	200264	300270	600264	300273	275	600288	500293	300277	400268	300278	400283	110272	300283	200272	400279	200265	500267
FA-EEF-2011-M	110274	100275	300296	271	100282	220274	110278	120283	110275	110280	210286	110281	110280	130285	100286	600293	120267	110277	300281	120276	320280	277	200287	310294	20264
JNS-CEJXXIII-2011-N	304	272	285	300	333	289	297	280	274	283	295	327	298	291	301	285	300	297	311	303	292	304	300	320	292
CM-CEDB-2010-N	381	359	349	390	389	329	340	362	350	383	356	357	356	351	360	339	355	356	355	348	348	336	339	356	351
JNS-CEJXXIII-2011-M	385	387	381	357	411	396	427	386	384	402	377	398	377	402	381	389	388	396	375	406	396	374	401	385	411
JNS-CEJXXIII-2011-V	397	382	399	397	383	410	401	397	377	395	401	366	409	409	416	396	393	383	411	366	381	407	405	393	409
JNS-CEDPII-2011-V	532	548	528	541	553	552	520	526	531	519	569	541	534	532	560	536	516	528	524	535	535	542	538	544	508
CM-CECM-2011-N	700708	300718	800733	710742	620728	500726	300700	400727	510722	600692	600728	800728	300741	300709	300706	700708	800745	410724	610732	500706	310720	500707	300723	700693	500733
JNS-CEDPII-2011-M	559	598	552	562	585	572	574	560	567	555	596	567	598	554	586	564	570	562	560	577	579	568	539	561	576
CL-CECL-2011-N-A	714	693	701	719	697	705	686	691	720	707	696	701	715	732	670	687	700	701	691	706	739	720	716	703	699
MGA-CEVB-2011-V	100591	300607	594	110586	10584	100589	100602	200611	100595	100598	100581	120598	602	100605	594	596	10589	210608	593	100604	100598	592	615	100610	100613
MGA-CEVB-2011-M	666	670	659	200672	100672	644	673	658	652	662	641	100669	646	670	695	661	679	100657	10643	628	645	100661	671	653	659
CL-CEASD-2008-V-A	821	807	795	100799	821	781	815	796	787	787	100777	774	100780	784	808	100804	782	778	817	811	806	803	794	767	791
CL-CEASD-2008-V-B	786	774	836	812	813	789	793	805	803	805	100800	797	774	820	817	787	802	799	791	816	773	769	797	790	787
MGA-CEDC-2011-V	500816	420797	200785	400792	310805	410798	600823	200775	120792	200808	500811	520814	300800	600797	400812	400800	500801	410807	600807	400804	110792	400803	500815	600819	500805
CL-CECL-2011-M-A	897	858	844	877	855	859	865	889	838	859	833	833	852	840	875	893	852	862	845	877	842	834	829	841	880
CL-CECL-2011-M-B	847	839	877	840	835	893	850	822	866	884	828	828	878	841	832	859	859	852	857	930	884	861	834	845	893
CM-CECM-2011-V	200917	200881	110889	100905	100911	10899	400905	310880	710893	110899	10887	200908	520900	300902	400916	200903	400945	100885	100888	200901	410898	300897	700916	100900	200888
CL-CECL-2011-V-A	921	906	905	883	877	892	910	853	916	877	897	894	898	908	870	901	875	898	877	922	891	890	897	887	890
CM-CEUP-2008-V	101124	301113	101127	401112	201065	101107	301135	201117	401099	301116	301096	301115	1120	501173	201085	201079	401134	201097	1129	301141	201091	601109	401120	101093	301094
CM-CEUP-2011-M	1.30114e+006	601119	501105	601125	801108	501125	701129	621127	1.10115e+006	1.10112e+006	801099	711126	511109	801112	411134	601107	911126	801103	301121	701128	621115	901094	601095	601136	801121
CM-CEUP-2011-V	511034	501040	711030	611063	501039	701051	211026	601048	411030	611072	411046	421006	211006	431018	311030	411030	401039	411017	421033	821039	401037	301014	421046	311036	201021
MGA-CEJXXIII-2010-V	101048	11057	201082	201016	101040	401089	11048	1034	101084	101049	101022	301083	401057	1051	501112	101054	1066	301043	1060	101083	101084	101022	1069	1017	211083
NE-CESVP-2011-V-A	1179	1179	1172	1171	1202	1194	1164	1177	1145	1208	1199	1185	1179	1189	1164	1166	1213	1173	1172	1203	1187	1189	1191	1172	1195
NE-CESVP-2011-V-B	1155	1218	1142	1184	1140	1165	1228	1202	1152	1164	101169	1166	1180	1167	101177	1156	1167	1194	1216	1200	1174	1174	1206	1196	1194
NE-CESVP-2011-V-C	1173	1208	1158	1166	101198	1175	1164	1170	1191	1180	1221	1172	1181	1166	1202	1167	1166	1190	1168	1176	1166	1187	1190	1181	1218
NE-CESVP-2011-M-A	1293	1324	1256	1293	1332	1313	1275	1330	1264	1286	1282	1289	1298	1278	1306	1308	1338	1276	1327	1330	1272	1284	1301	1283	1299
NE-CESVP-2011-M-B	1295	1304	1304	1317	1326	1266	1277	1280	1300	1265	1283	1245	1280	1274	1311	1299	1314	1301	1280	1299	1241	1255	1277	1280	1286
NE-CESVP-2011-M-C	1265	1300	1292	1300	1310	1304	1266	1297	1304	1316	1290	1295	1308	1292	1302	1319	1288	101295	1327	1303	1306	1291	1315	1324	1306
NE-CESVP-2011-M-D	1377	1331	1296	1307	1282	1331	1345	1285	1303	1335	1321	1284	1272	1311	1296	1331	1327	1286	1295	1263	1294	1316	1298	1303	1304
MGA-CEDC-2011-M	401177	311165	201180	201151	101154	301170	901190	701174	411185	201164	301169	701190	901164	411139	501169	401162	321155	1.01118e+006	301148	601175	501176	701149	301144	101150	401177
CM-CECM-2011-M	101354	301378	101387	101398	101369	301353	311375	1393	1383	211405	101356	301404	121366	101365	211404	11396	221387	201394	101391	701406	221394	211374	301382	301375	311356
MGA-CEGV-2011-M	102098	112089	302100	102075	12128	212079	302077	212084	102055	202124	202069	122078	202061	302105	102064	302114	102118	202109	2109	102101	302084	12056	112107	202079	102087
MGA-CEGV-2011-V	302303	302282	212284	322272	612267	302250	312292	402252	702246	302291	702282	312291	502278	402295	302268	102258	302255	302252	302265	402265	502261	102256	302216	202261	402274

Table 4.12 shows the best results, average and percentage ARD obtained out of the 25 runs for this method. It could not produce feasible schedules for all instances; it failed on instances CM-CECM-2011-N, MGA-CEDC-2011-V, CM-CECM-2011-V, CM-CEUP-2011-M, CM-CEUP-2011-V, MGA-CEDC-2011-M, and MGA-CEGV-2011-V. This method had more trouble getting rid of the fourth hard constraint violations than the fifth hard constraint violations. It could successfully get rid of all third hard constraints violations. Instances CM-CECM-2011-N and CM-CEUP-2011-M had the most number of hard constraint violations which is three. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances that produced feasible schedules is 9.30882% for the instance JNS-CEJXXIII-2011-M, and for all instances including the ones that produced at least one unfeasible solution is 113600% for the instance CM-CEUP-2011-N.

4.9 Results of the Swap Three Requirements (S3R) Method

Table 4.13 shows all results obtained by running the method S3R 25 times for each instance. This method obtained unfeasible solutions for instances MGA-CEDC-2011-V, CM-CEUP-2011-M and MGA-CEGV-2011-V. However, there was at most one hard constraint that was not satisfied in all these schedules. Furthermore, there were 31 instances in which every trial resulted in a feasible schedule (e.g. CM-CEUP-2011-N). Instances JNS-CEJXXIII-2011-N and CM-CEDB-2010-N got 25 best solutions. The gray cells show the best value found for that instance and the red cells show values for unfeasible solutions.

Table 4.12: Best results, average and percentage ARD with its score for each constraint, out of the 25 runs of the method CC

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	17	15	243	275	312675	113600
FA-EEF-2011-M	0	0	0	13	24	234	271	161880	59634.3
JNS-CEJXXIII-2011-N	0	0	0	17	3	252	272	297.32	9.30882
CM-CEDB-2010-N	0	0	0	17	6	306	329	355.8	8.1459
JNS-CEJXXIII-2011-M	0	0	0	21	3	333	357	390.88	9.4902
JNS-CEJXXIII-2011-V	0	0	0	24	9	333	366	395.32	8.01093
JNS-CEDPII-2011-V	0	0	0	19	12	477	508	535.68	5.44882
CM-CECM-2011-N	0	300000	0	52	45	603	300700	523520	74.1004
JNS-CEDPII-2011-M	0	0	0	23	21	495	539	569.64	5.6846
CL-CECL-2011-N-A	0	0	0	31	18	621	670	704.36	5.12836
MGA-CEVB-2011-V	0	0	0	31	21	540	592	82998.2	13920
MGA-CEVB-2011-M	0	0	0	31	21	576	628	25060.2	3890.48
CL-CEASD-2008-V-A	0	0	0	41	24	702	767	16795.4	2089.75
CL-CEASD-2008-V-B	0	0	0	46	30	693	769	4797.4	523.849
MGA-CEDC-2011-V	0	100000	10000	51	21	720	110792	404803	265.372
CL-CECL-2011-M-A	0	0	0	52	21	756	829	857.16	3.39686
CL-CECL-2011-M-B	0	0	0	51	15	756	822	857.36	4.3017
CM-CECM-2011-V	0	0	10000	50	27	810	10887	256501	2256.03
CL-CECL-2011-V-A	0	0	0	67	12	774	853	893.4	4.73623
CM-CEUP-2008-V	0	0	0	88	33	999	1120	257112	22856.4
CM-CEUP-2011-M	0	300000	0	98	60	963	301121	716319	137.884
CM-CEUP-2011-V	0	200000	0	55	57	909	201021	450234	123.974
MGA-CEJXXIII-2010-V	0	0	0	84	24	909	1017	138258	13494.7
NE-CESVP-2011-V-A	0	0	0	47	45	1053	1145	1182.72	3.29432
NE-CESVP-2011-V-B	0	0	0	48	39	1053	1140	9179.44	705.214
NE-CESVP-2011-V-C	0	0	0	57	57	1044	1158	5181.36	347.44
NE-CESVP-2011-M-A	0	0	0	71	42	1143	1256	1297.48	3.30255
NE-CESVP-2011-M-B	0	0	0	59	30	1152	1241	1286.36	3.65512
NE-CESVP-2011-M-C	0	0	0	68	36	1161	1265	5300.6	319.02
NE-CESVP-2011-M-D	0	0	0	69	33	1161	1263	1307.72	3.54078
MGA-CEDC-2011-M	0	100000	0	82	33	1035	101150	447566	342.478
CM-CECM-2011-M	0	0	0	105	63	1215	1383	198182	14229.8
MGA-CEGV-2011-M	0	0	0	141	78	1890	2109	161290	7547.7
MGA-CEGV-2011-V	0	100000	0	135	69	2052	102256	356669	248.8

Table 4.13: Results for the 25 of the method S3R. Gray cells represent the best solutions; red cells represent unfeasible solutions

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	269	270	269	269	270	269	269	269	269	269	269	269	269	270	269	270	269	269	269	269	269	269	269	270	269
FA-EEF-2011-M	255	257	255	255	255	255	255	256	255	255	255	255	255	255	255	255	255	255	255	256	255	255	255	255	256
JNS-CEJXXIII-2011-N	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298
JNS-CEJXXIII-2011-M	320	320	320	322	321	319	319	319	320	319	321	319	319	319	319	323	320	323	321	319	321	319	319	323	320
JNS-CEJXXIII-2011-V	324	325	324	323	323	323	324	324	323	324	329	323	323	323	333	323	324	323	327	325	323	324	323	324	324
JNS-CEDPII-2011-V	461	466	467	460	465	461	460	457	470	463	469	461	462	461	468	462	467	465	460	462	462	461	460	459	462
CM-CECM-2011-N	677	683	682	674	671	687	684	685	673	686	671	672	677	676	681	680	682	688	679	675	682	679	674	685	680
JNS-CEDPII-2011-M	494	487	487	491	489	486	485	491	489	484	485	493	491	493	491	488	492	493	486	490	491	494	491	482	490
CL-CECL-2011-N-A	631	630	635	634	635	634	631	629	627	631	628	630	629	632	633	630	628	631	632	630	631	627	632	632	633
MGA-CEVB-2011-V	558	559	564	562	561	564	558	563	555	562	562	559	557	566	562	562	558	557	564	567	564	560	565	556	560
MGA-CEVB-2011-M	601	595	584	578	577	593	586	589	585	582	587	577	585	602	577	586	587	598	580	597	583	593	591	577	596
CL-CEASD-2008-V-A	708	715	706	713	714	715	722	714	713	711	712	717	719	717	724	725	720	707	709	721	708	713	719	715	707
CL-CEASD-2008-V-B	708	713	706	719	715	712	719	719	721	701	712	716	718	717	715	717	715	709	714	717	714	712	713	719	715
MGA-CEDC-2011-V	738	739	739	739	738	733	743	740	738	740	741	745	743	746	741	100748	739	741	100741	739	741	742	742	736	737
CL-CECL-2011-M-A	763	764	753	754	762	753	758	762	758	760	759	765	758	757	771	763	759	754	760	754	757	754	757	756	764
CL-CECL-2011-M-B	761	746	749	746	749	747	759	764	753	745	750	753	756	755	752	753	755	756	749	751	766	752	753	748	754
CM-CECM-2011-V	829	832	823	832	836	834	835	838	834	827	829	844	829	823	831	824	832	837	824	816	828	829	829	826	836
CL-CECL-2011-V-A	802	793	790	788	801	789	808	792	788	784	791	777	802	794	797	792	792	787	783	790	794	788	801	788	795
CM-CEUP-2008-V	1020	1016	1018	1033	1005	1025	1018	1012	1014	1009	1020	1004	1019	1004	1022	1020	1037	1010	1023	1030	1005	1022	1013	1022	1011
CM-CEUP-2011-M	1056	1055	1054	1062	1055	1052	1056	1055	1054	1065	1061	1045	1048	1069	1056	1061	1058	11052	1061	1052	1057	1049	1057	1058	1064
CM-CEUP-2011-V	961	960	961	968	950	961	965	965	968	958	954	961	971	961	960	963	961	958	963	970	970	956	958	962	970
MGA-CEJXXIII-2010-V	949	960	970	954	953	955	974	950	956	955	946	952	974	949	960	954	945	971	961	954	960	957	954	959	943
NE-CESVP-2011-V-A	1053	1053	1045	1064	1052	1054	1062	1063	1046	1057	1040	1048	1062	1047	1056	1055	1051	1065	1066	1061	1064	1048	1048	1055	1056
NE-CESVP-2011-V-B	1054	1047	1049	1053	1047	1043	1053	1050	1065	1061	1059	1050	1052	1066	1063	1047	1058	1065	1047	1077	1064	1052	1048	1060	1066
NE-CESVP-2011-V-C	1048	1049	1051	1071	1050	1050	1057	1052	1052	1044	1046	1054	1053	1045	1038	1038	1042	1056	1065	1045	1049	1051	1054	1052	1059
NE-CESVP-2011-M-A	1166	1170	1171	1173	1161	1167	1172	1172	1170	1174	1159	1162	1166	1182	1164	1164	1185	1168	1171	1168	1157	1163	1167	1183	1158
NE-CESVP-2011-M-B	1164	1173	1172	1151	1166	1175	1165	1169	1159	1167	1165	1173	1174	1155	1176	1161	1162	1143	1170	1175	1169	1156	1165	1169	1175
NE-CESVP-2011-M-C	1189	1189	1180	1197	1187	1188	1164	1171	1197	1192	1182	1172	1164	1167	1174	1176	1178	1199	1203	1170	1161	1173	1194	1182	1171
NE-CESVP-2011-M-D	1186	1186	1173	1180	1172	1199	1176	1167	1191	1190	1179	1188	1210	1188	1179	1169	1180	1189	1184	1159	1180	1171	1173	1180	1203
MGA-CEDC-2011-M	1091	1086	1082	1103	1082	1088	1098	1090	1100	1098	1089	1095	1097	1104	1091	1102	1085	1092	1091	1094	1093	1084	1100	1088	1102
CM-CECM-2011-M	1299	1281	1290	1289	1288	1315	1295	1287	1298	1272	1281	1285	1279	1289	1283	1285	1284	1287	1294	1288	1282	1296	1292	1283	1288
MGA-CEGV-2011-M	1967	1957	1955	1968	1963	1981	1957	1954	1966	1971	1975	1964	1949	1992	1964	1987	1986	1936	1959	1980	1992	1963	1951	1937	1970
MGA-CEGV-2011-V	102138	2150	2160	2138	2143	2157	2148	2140	2143	2137	2141	2178	2144	2134	2130	2152	2173	2132	2135	2151	2136	2165	2128	2174	2140

48

Table 4.14 shows the best results obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances that produced feasible schedules is 2.00875% for the instance NE-CESVP-2011-M-B, and for all instances including the ones that produced at least one unfeasible solution is 1092.41% for the instance MGA-CEDC-2011-V. Instances JNS-CEJXXIII-2011-N and CM-CEDB-2010-N got a percentage ARD of 0.

4.10 Results of the Simulated Annealing (SA) Cooling Scheme

The problem with using this cooling scheme is that there are variables to be tuned such as the initial temperature t and the cooling rate α. Furthermore, this cooling scheme (i.e. $t = \alpha \times t$) can be placed at the end of each iteration: SAO, or in the local search: SAI, and it can give different results. It was placed in the local search for temperatures $t = 1$ and $t = 100000$, but it was only placed at the end of each iteration for temperature $t = 1$. A combination of one of these variations and the 2TQ perturbation was also implemented.

The first variation is when $t = 1$ and the cooling scheme is placed at the end of each iteration. This value for the temperature is used since the lowest value of all the soft constraints is 1. Except for $\alpha = 0.1$ and $\alpha = 0.4$ and $\alpha = 0.7$, the rest of variations did not achieve the best solution for the second instance which is the second smallest. As seen in Table 4.15, at least one of the variations found at least one unfeasible solution (i.e. its % ARD is big) for seven instances CM-CECM-2011-N, MGA-CEDC-2011-V, CM-CEUP-2011-M, CM-CEUP-2011-V, CM-CECM-2011-M, MGA-CEGV-2011-M, and MGA-CEGV-2011-V. The parameter that performed the best was $\alpha = 0.875$ in finding near-optimum solutions and $\alpha = 0.55$ in the similarity of its solution scores. All variations improved over Saviniec et al.'s % ARD at least in one instance. Gray cells represent the best solutions, cyan cells represent better % ARD than Saviniec et al.'s, bold numbers are the lowest percentage ARD of our methods obtained for that instance, and an asterisk (*) means that the solution is the same as Saviniec's (which is our lower bound).

The second variation is when $t = 1$ and the cooling scheme is placed in the local search. As seen in Table 4.16, There were two instances were at least one method found an unfeasible solution: CM-CECM-2011-N and MGA-CEDC-2011-V. The parameter that performed the best was $\alpha = 0.5$ in finding near-optimum solutions and $\alpha = 0.875$ in the similarity of their solution scores. Furthermore, since the best of this variation performed better than the best for the previous variation, this variation was chosen to be combined with the 2TQ perturbation method. There are ten instances where our % ARD is less than Saviniec et al.'s ILS-TQ. Instance CM-CEUP-2011-N's % ARD was surpassed by all variations, and these improved over Saviniec's % ARD at least once. Gray cells represent the best solutions, cyan cells represent better % ARD than Saviniec et al.'s, bold numbers

Table 4.14: Best results, average and percentage ARD with its score for each constraint, out of the 25 runs of the method S3R

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	269.2	0.0743494
FA-EEF-2011-M	0	0	0	6	15	234	255	255.2	0.0784314
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254	0
CM-CEDB-2010-N	0	0	0	10	0	288	298	298	0
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	320.16	0.363636
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	324.32	0.408669
JNS-CEDPII-2011-V	0	0	0	7	0	450	457	462.84	1.2779
CM-CECM-2011-N	0	0	0	32	45	594	671	679.32	1.23994
JNS-CEDPII-2011-M	0	0	0	11	3	468	482	489.32	1.51867
CL-CECL-2011-N-A	0	0	0	18	6	603	627	631	0.637959
MGA-CEVB-2011-V	0	0	0	15	0	540	555	561	1.08108
MGA-CEVB-2011-M	0	0	0	16	3	558	577	587.44	1.80936
CL-CEASD-2008-V-A	0	0	0	22	18	666	706	714.56	1.21246
CL-CEASD-2008-V-B	0	0	0	20	15	666	701	714.24	1.88873
MGA-CEDC-2011-V	0	0	0	31	18	684	733	8740.36	1092.41
CL-CECL-2011-M-A	0	0	0	27	15	711	753	759	0.796813
CL-CECL-2011-M-B	0	0	0	25	9	711	745	752.88	1.05772
CM-CECM-2011-V	0	0	0	33	18	765	816	830.28	1.75
CL-CECL-2011-V-A	0	0	0	30	0	747	777	792.24	1.96139
CM-CEUP-2008-V	0	0	0	59	27	918	1004	1017.28	1.32271
CM-CEUP-2011-M	0	0	0	58	42	945	1045	1456.48	39.3761
CM-CEUP-2011-V	0	0	0	32	27	891	950	962.2	1.28421
MGA-CEJXXIII-2010-V	0	0	0	52	18	873	943	956.6	1.44221
NE-CESVP-2011-V-A	0	0	0	38	30	972	1040	1054.84	1.42692
NE-CESVP-2011-V-B	0	0	0	38	33	972	1043	1055.84	1.23106
NE-CESVP-2011-V-C	0	0	0	39	36	963	1038	1050.84	1.23699
NE-CESVP-2011-M-A	0	0	0	44	42	1071	1157	1168.52	0.995678
NE-CESVP-2011-M-B	0	0	0	51	21	1071	1143	1165.96	2.00875
NE-CESVP-2011-M-C	0	0	0	51	21	1089	1161	1180.8	1.70543
NE-CESVP-2011-M-D	0	0	0	46	24	1089	1159	1182.08	1.99137
MGA-CEDC-2011-M	0	0	0	50	24	1008	1082	1093	1.01664
CM-CECM-2011-M	0	0	0	69	33	1170	1272	1288.4	1.28931
MGA-CEGV-2011-M	0	0	0	103	33	1800	1936	1965.76	1.53719
MGA-CEGV-2011-V	0	0	0	100	39	1989	2128	6146.68	188.848

Table 4.15: Best result and ARD comparison of the SA cooling scheme at the end of each iteration with $t = 1$ and for different values of α, including Saviniec et al.'s implementation. An asterisk (*) means that the value of the solution is the same as Saviniec et al's. Gray cells represent the best solution. Cyan cells represent better % ARD than Saviniec et al.'s and bold numbers represent the lowest % ARD of our methods.

Instance	Saviniec's ILS-TQ		SAO cooling scheme											
			0.1		0.4		0.5		0.55		0.7		0.875	
	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD
CM-CEUP-2011-N	269	0.18	*	0.21	*	0.13	*	0.13	*	0.19	*	0.1	*	**0.09**
FA-EEF-2011-M	254	0.38	*	0.61	*	0.43	255	0.19	255	0.28	*	0.5	255	**0.11**
JNS-CEJXXIII-2011-N	254	0	*	**0**	*	**0**	*	**0**	*	**0**	*	**0**	*	**0**
CM-CEDB-2010-N	298	0	*	0.09	*	0.12	*	0.07	*	**0.04**	*	0.11	*	0.05
JNS-CEJXXIII-2011-M	319	0.01	*	**0.38**	*	0.49	*	0.59	*	0.41	*	0.39	*	0.5
JNS-CEJXXIII-2011-V	323	0.09	*	0.51	*	0.28	*	0.31	*	0.3	*	0.32	*	**0.25**
JNS-CEDPII-2011-V	457	0.76	459	1.03	460	0.97	460	**0.84**	*	1.74	460	1.05	460	0.97
CM-CECM-2011-N	667	1.65	672	1.74	673	1.54	670	1.96	669	1.99	670	1195.58	671	**1.19**
JNS-CEDPII-2011-M	481	0.84	483	1.31	482	1.62	482	**1.05**	*	1.83	483	1.37	483	1.16
CL-CECL-2011-N-A	627	0.06	*	0.66	*	0.6	*	0.66	*	**0.52**	*	0.59	*	**0.52**
MGA-CEVB-2011-V	552	0.54	556	**0.87**	554	1.16	554	0.93	554	1.01	555	0.94	554	0.95
MGA-CEVB-2011-M	571	1.37	574	2.17	576	1.69	576	1.95	575	**1.61**	577	1.62	575	2.11
CL-CEASD-2008-V-A	699	0.57	705	1.15	708	**1.08**	706	1.22	705	1.14	705	1.29	705	1.17
CL-CEASD-2008-V-B	699	0.82	703	1.61	708	**0.9**	705	1.46	704	1.32	707	1.08	706	1.24
MGA-CEDC-2011-V	728	0.47	733	2729.4	733	**1637.96**	730	3836.73	729	2196.15	727	1652.32	728	4946.66
CL-CECL-2011-M-A	744	0.31	751	**0.75**	751	0.91	750	1.02	751	0.9	751	1.08	750	0.81
CL-CECL-2011-M-B	736	0.44	745	0.99	742	1.19	744	**0.78**	744	1.06	743	1.02	740	1.43
CM-CECM-2011-V	806	1.49	816	1.76	818	1.34	819	1.25	815	1.55	818	**1.19**	817	1.42
CL-CECL-2011-V-A	772	0.31	784	**0.79**	777	1.51	780	1.06	781	1.05	779	1.29	779	1.14
CM-CEUP-2008-V	985	0.69	998	1.88	991	2.31	1003	1.31	1003	**1.28**	1002	1.71	992	2.16
CM-CEUP-2011-M	1028	0.83	1044	1150.23	1043	0.88	1042	1536.49	1041	769.41	1045	**0.82**	1044	1188.52
CM-CEUP-2011-V	940	0.98	945	1.95	952	**1.09**	950	422.24	947	1.77	951	421.71	952	841.35
MGA-CEJXXIII-2010-V	923	1.39	940	1.24	935	1.97	944	1.19	942	1.41	935	2.2	947	**1.05**
NE-CESVP-2011-V-A	1027	0.62	1042	1.12	1043	1.28	1043	**1**	1036	1.47	1043	1.13	1037	1.4
NE-CESVP-2011-V-B	1028	0.86	1045	0.91	1044	0.91	1037	1.88	1045	**0.86**	1040	1.25	1045	0.98
NE-CESVP-2011-V-C	1020	0.83	1034	1.28	1033	1.29	1037	**0.94**	1034	1.23	1039	1.33	1029	1.88
NE-CESVP-2011-M-A	1130	0.66	1154	0.92	1155	0.91	1155	1.03	1157	0.88	1158	**0.79**	1150	1.33
NE-CESVP-2011-M-B	1121	0.88	1145	1.22	1145	1.3	1140	1.63	1147	1.42	1136	2.13	1147	**1.2**
NE-CESVP-2011-M-C	1141	0.8	1160	1.54	1164	1.1	1165	**1**	1163	1.5	1162	1.39	1161	1.49
NE-CESVP-2011-M-D	1138	0.77	1152	2.26	1163	1.09	1152	2.32	1166	**1.02**	1163	1.17	1161	1.38
MGA-CEDC-2011-M	1058	0.75	1075	1.1	1080	**0.74**	1076	1.21	1076	1.08	1077	1.12	1071	1.5
CM-CECM-2011-M	1243	0.93	1261	319.02	1270	1	1269	1.56	1269	**0.85**	1270	1.02	1276	0.87
MGA-CEGV-2011-M	1865	0.83	1929	1.71	1931	1.21	1928	1.32	1936	**0.96**	1936	42.29	1917	1.6
MGA-CEGV-2011-V	2037	1.13	2115	379.43	2100	382.7	2099	192.03	2104	**191.62**	2118	567.26	2093	192.81
Best count	34	26	14	5	11	6	10	7	14	**10**	11	4	15	8

Table 4.16: Best result and ARD comparison of the SA cooling scheme within the local search with $t = 1$ and for different values of α, including Saviniec et al.'s implementation. An asterisk (*) means that the value of the solution is the same as Saviniec et al's. Gray cells represent the best solution. Cyan cells represent better % ARD than Saviniec et al.'s, and bold numbers represent the lowest % ARD of our methods.

Instance	Saviniec's ILS-TQ		SAI cooling scheme 0.1		0.5		0.75		0.875		0.9375	
	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD
CM-CEUP-2011-N	269	0.18	*	**0.09**	*	0.1	*	**0.09**	*	**0.09**	*	0.15
FA-EEF-2011-M	254	0.38	*	0.47	*	0.52	*	**0.39**	*	0.52	*	0.52
JNS-CEJXXIII-2011-N	254	0	*	**0**	*	**0**	*	**0**	*	0.02	*	0.09
CM-CEDB-2010-N	298	0	*	0.07	*	**0**	*	0.07	*	0.19	*	0.39
JNS-CEJXXIII-2011-M	319	0.01	*	0.33	*	**0.23**	*	0.4	*	0.3	*	0.6
JNS-CEJXXIII-2011-V	323	0.09	*	0.27	*	**0.1**	*	0.3	*	0.56	*	0.53
JNS-CEDPII-2011-V	457	0.76	461	**0.58**	461	0.82	*	1.32	461	0.98	461	1.32
CM-CECM-2011-N	667	1.65	669	1.36	668	1.62	670	598.21	672	**1.36**	668	1.53
JNS-CEDPII-2011-M	481	0.84	482	1.15	483	**0.73**	*	1.22	483	1.21	484	1.91
CL-CECL-2011-N-A	627	0.06	*	0.61	*	0.67	*	0.74	629	**0.61**	630	0.95
MGA-CEVB-2011-V	552	0.54	555	0.89	555	0.87	553	1.14	558	**0.85**	560	1.51
MGA-CEVB-2011-M	571	1.37	578	**1.06**	573	2.07	573	1.76	575	1.8	580	2.06
CL-CEASD-2008-V-A	699	0.57	707	**0.71**	705	0.91	704	1.31	710	1.27	714	1.32
CL-CEASD-2008-V-B	699	0.82	705	1.29	706	1.07	707	**0.88**	709	1.16	714	1.34
MGA-CEDC-2011-V	728	0.47	730	0.85	730	1096.85	730	549.07	731	1.38	739	**0.64**
CL-CECL-2011-M-A	744	0.31	750	1.01	751	0.97	751	0.92	754	**0.86**	759	1.08
CL-CECL-2011-M-B	736	0.44	745	**0.71**	740	1.28	743	0.79	745	1.22	747	1.52
CM-CECM-2011-V	806	1.49	816	1.42	814	1.7	817	**1.26**	818	1.3	820	1.59
CL-CECL-2011-V-A	772	0.31	779	1.11	778	1.25	782	0.98	788	1.04	796	**0.94**
CM-CEUP-2008-V	985	0.69	981	2.92	985	2.29	996	**1.18**	999	1.64	1003	1.91
CM-CEUP-2011-M	1028	0.83	1040	1.03	1036	1.14	1038	1.29	1049	**0.85**	1046	1.75
CM-CEUP-2011-V	940	0.98	948	1.24	944	1.7	950	**1.04**	957	1.07	959	1.48
MGA-CEJXXIII-2010-V	923	1.39	935	1.95	941	1.44	935	1.53	946	**0.99**	953	1.31
NE-CESVP-2011-V-A	1027	0.62	1040	1.03	1033	1.35	1032	1.46	1043	**0.76**	1037	1.84
NE-CESVP-2011-V-B	1028	0.86	1038	1.31	1039	1.3	1040	1.08	1046	**0.79**	1047	1.22
NE-CESVP-2011-V-C	1020	0.83	1031	1.09	1029	1.56	1032	**0.97**	1030	1.44	1038	1.11
NE-CESVP-2011-M-A	1130	0.66	1143	1.53	1149	**0.74**	1147	0.89	1146	1.52	1156	1.28
NE-CESVP-2011-M-B	1121	0.88	1142	1.14	1140	1.28	1141	1.04	1144	1.12	1155	**0.86**
NE-CESVP-2011-M-C	1141	0.8	1161	1.26	1155	1.37	1157	0.97	1164	**0.95**	1161	1.71
NE-CESVP-2011-M-D	1138	0.77	1156	1.35	1153	1.52	1159	**0.81**	1153	1.87	1165	1.38
MGA-CEDC-2011-M	1058	0.75	1076	1.03	1073	1.08	1073	1.09	1086	**0.85**	1094	1.4
CM-CECM-2011-M	1243	0.93	1258	1.3	1258	**1.24**	1256	1.36	1263	1.28	1270	1.4
MGA-CEGV-2011-M	1865	0.83	1900	2.12	1913	**1.12**	1905	1.24	1908	1.14	1921	1.35
MGA-CEGV-2011-V	2037	1.13	2075	1.88	2081	**1.07**	2077	1.2	2085	1.22	2105	1.11
Best count	34	24	16	6	20	9	17	9	7	**11**	7	3

are the lowest percentage ARD of our methods obtained for that instance, and an asterisk (*) means that the solution is the same as Saviniec's (which is our lower bound).

The next variation is when $t = 100000$ and the cooling scheme is placed in the local search. This temperature was chosen because that is the highest value a hard constraint violation can take. As seen in Table 4.17, this variation gave more consistent solutions than the previous variations (e.g. except for $\alpha = 0.025$ and $\alpha = 0.5$, there is no instance with at least one unfeasible solution). When $\alpha = 0.5$ and $t = 1$ for feasible solutions, it found 34 best solutions compared to the rest, except Saviniec's, and found no unfeasible solutions. However, its solutions were not as consistent as $\alpha = 0.025$. All variations got at least one better % ARD over Saviniec's. Gray cells represent the best solutions, cyan cells represent better % ARD than Saviniec et al.'s, bold numbers are the lowest percentage ARD of our methods obtained for that instance, and an asterisk (*) means that the solution is the same as Saviniec's (which is our lower bound).

Table 4.17: Best result and ARD comparison of the SA cooling scheme within the local search with $t = 100000$ and for different values of α, including Saviniec et al.'s implementation. An asterisk (*) means that the value of the solution is the same as Saviniec et al's. The $t = 1$ means that temperature was used when a feasible solution was found. Gray cells represent the best solution. Cyan cells represent better % ARD than Saviniec et al.'s, and bold numbers represent the lowest % ARD of our methods.

Instance	Saviniec's ILS-TQ		SAI cooling scheme											
			0.5 and $t = 1$		0.025		0.05		0.1		0.25		0.5	
	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD
CM-CEUP-2011-N	269	0.18	*	0.65	*	**0.07**	*	0.1	*	0.09	*	0.25	271	1.51
FA-EEF-2011-M	254	0.38	255	0.96	256	1.13	255	1.51	257	**0.79**	258	0.85	261	12.48
JNS-CEJXXIII-2011-N	254	0	*	**0**	255	1.87	257	1.56	257	1.85	260	3.85	499	3.24
CM-CEDB-2010-N	298	0	*	**0.07**	304	2.25	306	1.93	305	3.23	308	3.87	618	1.61
JNS-CEJXXIII-2011-M	319	0.01	*	**0.49**	332	2.37	333	2.26	332	3.29	342	2.76	689	2.09
JNS-CEJXXIII-2011-V	323	0.09	*	**0.45**	331	2.78	336	1.96	332	3.49	342	2.39	700	1.13
JNS-CEDPII-2011-V	457	0.76	458	1.6	483	**0.97**	479	1.74	483	1.05	486	1.48	508	67.91
CM-CECM-2011-N	667	1.65	689	2.17	708	1.69	702	1.99	704	**1.22**	699	1.32	705	2.14
JNS-CEDPII-2011-M	481	0.84	483	1.69	509	**1.67**	509	2.04	508	2.14	514	2.3	559	41.99
CL-CECL-2011-N-A	627	0.06	628	**0.45**	646	2.48	653	1.39	652	1.67	663	1.92	711	70.44
MGA-CEVB-2011-V	552	0.54	559	1.07	574	0.59	575	**0.5**	574	0.68	575	0.97	683	0.56
MGA-CEVB-2011-M	571	1.37	579	1.73	609	1.24	611	1.18	611	1.53	616	1.4	838	**0.9**
CL-CEASD-2008-V-A	699	0.57	711	1.05	745	**0.96**	742	1.45	743	1.67	745	1.56	779	37.6
CL-CEASD-2008-V-B	699	0.82	710	**0.9**	741	1.54	745	1.18	739	2	752	0.98	771	42.39
MGA-CEDC-2011-V	728	0.47	739	1.36	761	1.5	762	1.34	762	1.12	758	**0.92**	760	1.79
CL-CECL-2011-M-A	744	0.31	750	**0.78**	799	1.31	794	1.68	801	1.21	809	1.31	835	68.07
CL-CECL-2011-M-B	736	0.44	742	**1.13**	792	1.55	790	1.63	792	1.79	801	2.23	1498	1.41
CM-CECM-2011-V	806	1.49	825	2.27	853	1.06	854	**0.96**	851	1.04	849	1.49	860	7.59
CL-CECL-2011-V-A	772	0.31	775	1.58	835	**0.88**	830	1.51	828	1.66	837	1.7	1466	1.75
CM-CEUP-2008-V	985	0.69	1013	1.84	1055	1.4	1058	1.55	1060	1.12	1066	**0.68**	1081	10.86
CM-CEUP-2011-M	1028	0.83	1065	1.96	1100	873.57	1085	1.96	1091	0.94	1090	**0.65**	1091	0.86
CM-CEUP-2011-V	940	0.98	965	1.48	993	1.24	992	**0.88**	991	0.89	985	1.32	997	1.04
MGA-CEJXXIII-2010-V	923	1.39	948	2.87	994	1.32	1004	**0.65**	991	1.69	992	1.7	1018	15.61
NE-CESVP-2011-V-A	1027	0.62	1039	1.54	1109	1.22	1113	**1.08**	1093	2.73	1110	1.19	1879	1.51
NE-CESVP-2011-V-B	1028	0.86	1041	1.63	1107	**0.93**	1103	1.66	1104	1.6	1111	1.4	1135	51.87
NE-CESVP-2011-V-C	1020	0.83	1032	1.82	1100	**0.99**	1105	1.11	1102	1.06	1101	1.65	1135	52.94
NE-CESVP-2011-M-A	1130	0.66	1149	1.19	1207	1.76	1219	1.05	1222	**1.04**	1220	1.4	1255	53.07
NE-CESVP-2011-M-B	1121	0.88	1144	1.29	1205	1.63	1207	1.89	1218	**0.8**	1207	1.89	1268	52.46
NE-CESVP-2011-M-C	1141	0.8	1162	1.27	1236	**0.52**	1227	1.53	1234	1.03	1232	1.68	1266	46.31
NE-CESVP-2011-M-D	1138	0.77	1161	1.44	1234	**0.69**	1230	1.14	1227	1.38	1230	1.57	1265	40.8
MGA-CEDC-2011-M	1058	0.75	1103	0.99	1122	0.94	1123	0.79	1122	**0.71**	1115	1.12	1134	1.11
CM-CECM-2011-M	1243	0.93	1290	1.58	1333	**0.68**	1328	1.23	1328	1.14	1323	1.39	1360	9.79
MGA-CEGV-2011-M	1865	0.83	1962	1.52	2027	**0.81**	2000	2.37	2019	1.03	2019	1.12	2079	18.82
MGA-CEGV-2011-V	2037	1.13	2137	1.94	2206	0.91	2211	0.7	2210	**0.6**	2205	0.76	2235	19.05
Best count	34	18	34	8	1	**11**	2	5	1	6	1	3	0	1

Table 4.18 shows all results obtained by running the method SAI, where temperature $t = 1$ and the cooling rate $\alpha = 0.5$, 25 times for each instance. This variation had no problems dealing with hard constraints in most of these instances. However, there was at most one hard constraint that was not satisfied in all these schedules. Furthermore, there were 33 instances in which every trial resulted in a feasible schedule (e.g. CM-CEUP-2011-N), except for instance MGA-CEDC-2011-V that got two unfeasible solutions. Instances JNS-CEJXXIII-2011-N and CM-CEDB-2010-N got 25 best solutions. The gray cells show the best value found for that instance whereas red cells show values for unfeasible solutions.

Table 4.18: Results for the 25 runs using the SAI cooling scheme where temperature $t = 1$ and cooling rate $\alpha = 0.5$. Gray cells represent the best solutions; red cells represent unfeasible solutions

Instance	Runs																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	269	269	270	270	269	269	269	269	270	269	269	270	269	270	269	269	269	270	269	269	269	269	270	269	269
FA-EEF-2011-M	256	255	254	255	255	254	257	255	255	255	256	255	255	255	255	256	255	256	255	255	255	257	257	255	255
JNS-CEJXXIII-2011-N	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298
JNS-CEJXXIII-2011-M	319	319	319	319	319	324	319	319	319	322	319	322	320	321	322	319	319	320	319	319	319	319	319	319	319
JNS-CEJXXIII-2011-V	325	323	323	324	324	324	323	323	323	323	323	323	323	323	323	323	323	323	323	323	323	325	324	323	323
JNS-CEDPII-2011-V	468	462	464	465	461	469	463	466	466	469	466	461	463	474	463	470	462	465	462	466	461	461	468	462	463
CM-CECM-2011-N	674	681	681	671	683	682	681	670	679	679	687	678	675	670	677	684	679	682	693	681	668	685	680	676	674
JNS-CEDPII-2011-M	492	488	483	485	485	494	486	483	487	485	485	483	489	490	483	484	486	486	489	486	490	490	484	485	485
CL-CECL-2011-N-A	631	630	630	631	629	632	630	631	631	633	633	630	633	628	634	634	627	636	629	630	634	633	634	629	628
MGA-CEVB-2011-V	557	561	562	559	564	555	556	558	558	558	560	561	560	559	561	560	561	560	557	565	560	564	561	558	561
MGA-CEVB-2011-M	583	582	580	595	585	601	580	588	575	580	575	589	593	576	596	577	580	582	594	573	599	575	592	593	578
CL-CEASD-2008-V-A	713	712	712	717	716	714	710	714	709	706	713	708	707	711	714	705	708	717	706	712	709	709	710	711	722
CL-CEASD-2008-V-B	716	712	712	713	714	718	715	716	713	711	715	707	716	720	714	710	715	713	716	712	715	706	712	712	715
MGA-CEDC-2011-V	736	739	100730	742	737	735	743	743	734	738	737	738	733	100737	743	734	741	739	730	734	737	740	736	736	733
CL-CECL-2011-M-A	759	754	757	753	761	757	760	752	756	763	762	758	752	762	767	752	759	751	764	760	765	758	754	761	761
CL-CECL-2011-M-B	744	750	750	749	744	748	744	757	747	754	748	756	746	749	745	760	752	759	754	747	747	740	751	748	747
CM-CECM-2011-V	831	827	819	827	824	832	825	836	814	838	823	814	825	826	828	832	833	831	828	822	828	826	835	841	830
CL-CECL-2011-V-A	788	781	790	787	796	797	781	791	797	782	793	799	786	781	788	790	782	787	796	791	786	785	781	780	778
CM-CEUP-2008-V	1004	988	991	1014	1010	1012	1008	999	1020	996	1004	1006	1018	1022	993	1026	1000	1005	1012	985	1032	1017	1003	1005	1018
CM-CEUP-2011-M	1052	1055	1046	1057	1055	1054	1053	1044	1046	1055	1044	1040	1046	1042	1048	1042	1047	1046	1056	1044	1044	1036	1045	1050	1049
CM-CEUP-2011-V	969	972	944	948	962	963	953	957	960	951	961	964	962	961	954	962	957	962	965	955	956	964	966	963	971
MGA-CEJXXIII-2010-V	968	943	941	952	953	948	956	964	953	966	959	955	954	963	959	952	949	959	955	947	950	944	960	957	957
NE-CESVP-2011-V-A	1053	1043	1042	1039	1043	1041	1043	1057	1049	1044	1061	1041	1033	1050	1047	1040	1057	1057	1051	1047	1042	1047	1058	1047	1042
NE-CESVP-2011-V-B	1054	1043	1050	1045	1056	1039	1058	1059	1062	1053	1048	1053	1045	1066	1054	1057	1064	1058	1050	1049	1053	1055	1051	1042	1049
NE-CESVP-2011-V-C	1049	1066	1056	1035	1029	1041	1044	1045	1040	1047	1046	1043	1041	1036	1039	1043	1039	1050	1060	1046	1041	1043	1052	1057	1038
NE-CESVP-2011-M-A	1155	1157	1158	1164	1163	1164	1151	1149	1157	1152	1149	1157	1160	1150	1152	1155	1156	1152	1166	1158	1178	1152	1160	1168	1156
NE-CESVP-2011-M-B	1166	1161	1157	1151	1142	1147	1163	1151	1156	1172	1165	1150	1145	1142	1161	1145	1152	1140	1141	1162	1154	1170	1148	1164	1161
NE-CESVP-2011-M-C	1168	1168	1168	1180	1176	1174	1172	1155	1181	1186	1177	1167	1172	1169	1166	1184	1179	1175	1163	1173	1155	1157	1167	1159	1179
NE-CESVP-2011-M-D	1181	1171	1165	1162	1189	1170	1177	1177	1175	1165	1161	1179	1167	1162	1153	1173	1174	1180	1164	1165	1170	1159	1180	1174	1171
MGA-CEDC-2011-M	1078	1084	1086	1098	1084	1087	1081	1082	1073	1075	1083	1088	1089	1081	1074	1089	1083	1083	1090	1095	1090	1081	1086	1093	1081
CM-CECM-2011-M	1285	1269	1271	1288	1277	1263	1265	1281	1279	1265	1258	1282	1266	1271	1289	1264	1272	1273	1277	1271	1277	1273	1276	1284	1263
MGA-CEGV-2011-M	1940	1938	1917	1967	1933	1950	1936	1943	1954	1934	1915	1961	1945	1931	1922	1929	1925	1948	1913	1927	1917	1948	1913	1918	1938
MGA-CEGV-2011-V	2107	2108	2089	2113	2121	2104	2121	2104	2096	2093	2106	2111	2095	2082	2110	2105	2104	2110	2087	2097	2081	2103	2116	2113	2105

55

Table 4.19 shows the best results obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances that produced feasible schedules is 2.28629% for the instance CM-CEUP-2008-V, and for all instances including the ones that produced at least one unfeasible solution is 1096.85% for the instance MGA-CEDC-2011-V. Instances JNS-CEJXXIII-2011-N and CM-CEDB-2010-N got a percentage ARD of 0.

Table 4.20 shows all results obtained by running the method SA 25 times for each instance, where temperature $t = 100000$, the cooling rate $\alpha = 0.5$, and whenever a feasible solution is obtained, the temperature is set to 1. The most important thing to note is that it has no unfeasible solutions for all instances. Instance JNS-CEJXXIII-2011-N has 25 best solutions; however, the first two instances only have two best solutions at most. The gray cells show the best value found for that instance and red cells show values for unfeasible solutions.

Table 4.19: Best results, average and percentage ARD, with its score for each constraint, out of the 25 runs for the SAI method where $t = 1$ and $\alpha = 0.5$

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	269.28	0.104089
FA-EEF-2011-M	0	0	0	5	15	234	254	255.32	0.519685
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254	0
CM-CEDB-2010-N	0	0	0	10	0	288	298	298	0
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	319.72	0.225705
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	323.32	0.0990712
JNS-CEDPII-2011-V	0	0	0	8	3	450	461	464.8	0.824295
CM-CECM-2011-N	0	0	0	35	39	594	668	678.8	1.61677
JNS-CEDPII-2011-M	0	0	0	12	3	468	483	486.52	0.728778
CL-CECL-2011-N-A	0	0	0	18	6	603	627	631.2	0.669856
MGA-CEVB-2011-V	0	0	0	15	0	540	555	559.84	0.872072
MGA-CEVB-2011-M	0	0	0	15	0	558	573	584.84	2.06632
CL-CEASD-2008-V-A	0	0	0	21	18	666	705	711.4	0.907801
CL-CEASD-2008-V-B	0	0	0	22	18	666	706	713.52	1.06516
MGA-CEDC-2011-V	0	0	0	28	18	684	730	8737	1096.85
CL-CECL-2011-M-A	0	0	0	22	18	711	751	758.32	0.9747
CL-CECL-2011-M-B	0	0	0	20	9	711	740	749.44	1.27568
CM-CECM-2011-V	0	0	0	31	18	765	814	827.8	1.69533
CL-CECL-2011-V-A	0	0	0	31	0	747	778	787.72	1.24936
CM-CEUP-2008-V	0	0	0	52	24	909	985	1007.52	2.28629
CM-CEUP-2011-M	0	0	0	49	42	945	1036	1047.84	1.14286
CM-CEUP-2011-V	0	0	0	29	24	891	944	960.08	1.70339
MGA-CEJXXIII-2010-V	0	0	0	53	15	873	941	954.56	1.44102
NE-CESVP-2011-V-A	0	0	0	40	21	972	1033	1046.96	1.3514
NE-CESVP-2011-V-B	0	0	0	37	30	972	1039	1052.52	1.30125
NE-CESVP-2011-V-C	0	0	0	39	36	954	1029	1045.04	1.55879
NE-CESVP-2011-M-A	0	0	0	45	24	1080	1149	1157.56	0.744996
NE-CESVP-2011-M-B	0	0	0	51	27	1062	1140	1154.64	1.28421
NE-CESVP-2011-M-C	0	0	0	48	27	1080	1155	1170.8	1.36797
NE-CESVP-2011-M-D	0	0	0	49	24	1080	1153	1170.56	1.52298
MGA-CEDC-2011-M	0	0	0	44	21	1008	1073	1084.56	1.07735
CM-CECM-2011-M	0	0	0	61	36	1161	1258	1273.56	1.23688
MGA-CEGV-2011-M	0	0	0	116	33	1764	1913	1934.48	1.12284
MGA-CEGV-2011-V	0	0	0	104	33	1944	2081	2103.24	1.06872

58

Table 4.20: Results for the 25 runs using the SAI cooling scheme where temperature $t = 100000$, cooling rate $\alpha = 0.5$, and $t = 1$ when the solution is feasible. Gray cells represent the best solutions; red cells represent unfeasible solutions

	Runs																								
Instance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	271	271	272	271	271	270	271	271	271	271	271	271	271	271	271	270	271	271	271	271	269	271	271	269	270
FA-EEF-2011-M	257	255	259	257	259	258	256	257	257	257	260	257	259	259	256	259	256	256	258	257	257	256	257	260	257
JNS-CEJXXIII-2011-N	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	298	298	298	298	298	298	299	298	298	298	298	298	298	298	298	301	298	298	298	298	298	299	298	298
JNS-CEJXXIII-2011-M	321	319	320	321	321	320	320	319	319	321	320	321	319	320	324	321	320	323	325	322	319	321	320	319	319
JNS-CEJXXIII-2011-V	325	325	325	323	332	324	324	326	323	324	323	323	323	324	323	325	324	324	324	323	325	325	325	323	326
JNS-CEDPII-2011-V	462	468	469	472	458	467	467	466	465	465	467	463	462	461	464	472	464	465	465	462	469	462	466	466	466
CM-CECM-2011-N	711	705	707	708	705	702	704	701	709	701	717	707	707	697	701	689	695	703	705	707	694	719	700	704	701
JNS-CEDPII-2011-M	486	493	486	491	499	494	492	495	483	491	490	497	489	494	496	484	483	489	496	493	495	492	489	490	492
CL-CECL-2011-N-A	630	628	632	632	633	629	628	630	631	632	631	631	631	630	631	628	631	632	635	633	628	629	636	630	630
MGA-CEVB-2011-V	569	563	565	561	564	567	563	568	565	567	569	564	563	565	567	561	570	559	569	563	563	565	563	560	571
MGA-CEVB-2011-M	583	587	582	584	580	579	607	599	591	592	592	604	582	591	593	580	588	593	598	587	580	584	584	592	593
CL-CEASD-2008-V-A	725	712	720	715	711	715	717	731	714	722	723	722	717	719	718	724	711	725	711	718	723	711	720	719	719
CL-CEASD-2008-V-B	718	716	712	716	710	727	711	713	711	718	715	712	714	717	719	715	711	714	718	721	714	713	737	718	719
MGA-CEDC-2011-V	750	750	745	755	760	750	745	739	750	752	753	741	750	748	749	745	747	747	754	755	754	740	747	751	749
CL-CECL-2011-M-A	752	757	757	755	753	759	754	756	761	757	760	754	755	760	758	751	751	750	755	760	755	759	753	756	758
CL-CECL-2011-M-B	743	756	751	748	751	752	749	743	749	765	751	753	751	749	757	751	745	751	753	757	744	742	748	753	748
CM-CECM-2011-V	845	847	853	860	840	839	843	846	845	849	825	849	842	855	843	835	840	843	840	858	847	836	844	835	834
CL-CECL-2011-V-A	783	791	793	782	780	796	787	788	788	793	795	789	775	790	801	784	787	790	780	794	784	786	782	782	782
CM-CEUP-2008-V	1026	1053	1036	1025	1034	1013	1019	1032	1028	1025	1036	1031	1025	1045	1041	1036	1034	1034	1017	1025	1024	1020	1058	1043	1030
CM-CEUP-2011-M	1081	1065	1076	1101	1086	1090	1086	1080	1088	1082	1085	1093	1088	1094	1079	1075	1086	1095	1080	1085	1087	1088	1092	1086	1099
CM-CEUP-2011-V	970	995	993	974	969	978	988	980	980	979	978	988	987	985	974	967	976	978	977	983	971	983	981	983	965
MGA-CEJXXIII-2010-V	983	948	962	984	969	990	988	980	958	977	958	983	975	960	978	1001	980	980	977	973	975	977	980	974	970
NE-CESVP-2011-V-A	1062	1067	1052	1045	1070	1049	1060	1044	1051	1052	1058	1056	1071	1048	1064	1063	1050	1054	1047	1063	1045	1048	1050	1039	1066
NE-CESVP-2011-V-B	1058	1054	1062	1059	1063	1059	1055	1056	1056	1045	1058	1041	1069	1046	1059	1055	1077	1068	1059	1068	1061	1054	1053	1057	1057
NE-CESVP-2011-V-C	1053	1040	1051	1043	1032	1050	1056	1063	1040	1055	1061	1046	1060	1032	1049	1071	1068	1065	1046	1035	1046	1061	1051	1040	1055
NE-CESVP-2011-M-A	1154	1169	1172	1163	1169	1159	1171	1166	1153	1160	1174	1165	1159	1175	1155	1159	1164	1175	1149	1152	1156	1163	1161	1171	1154
NE-CESVP-2011-M-B	1151	1150	1161	1162	1157	1158	1154	1167	1177	1155	1148	1168	1164	1164	1168	1166	1156	1147	1148	1166	1144	1156	1161	1149	1173
NE-CESVP-2011-M-C	1172	1174	1181	1174	1168	1172	1181	1182	1183	1180	1178	1171	1162	1178	1185	1174	1169	1178	1182	1181	1178	1187	1174	1176	1179
NE-CESVP-2011-M-D	1172	1179	1166	1175	1177	1171	1185	1173	1191	1181	1184	1203	1175	1176	1164	1166	1195	1161	1180	1171	1175	1173	1195	1177	1178
MGA-CEDC-2011-M	1112	1105	1119	1113	1120	1121	1108	1108	1103	1108	1115	1116	1121	1115	1110	1112	1116	1115	1115	1107	1123	1125	1117	1112	1113
CM-CECM-2011-M	1304	1332	1308	1328	1314	1320	1315	1306	1296	1332	1290	1300	1309	1306	1299	1302	1316	1302	1315	1328	1293	1319	1306	1294	1326
MGA-CEGV-2011-M	1999	2025	1979	2004	1996	1992	1981	2009	2013	2005	1971	1962	2001	1987	1979	1969	1990	1974	1998	1970	1999	2014	1999	1979	1999
MGA-CEGV-2011-V	2164	2181	2175	2207	2185	2210	2149	2141	2187	2171	2178	2151	2191	2213	2137	2163	2214	2202	2193	2179	2142	2188	2182	2178	2178

Table 4.21 shows the best results obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances is 2.8692% for the instance MGA-CEJXXIII-2010-V. For the instance JNS-CEJXXIII-2011-N, it got a percentage ARD of 0.

We also tried to combine the 2TQ technique with the SA one where $t = 1$ and $\alpha = 0.5$. Table 4.22 shows all results obtained for this method 25 times for each instance. This method only obtained an unfeasible solution in the last instances MGA-CEGV-2011-V. However, there was at most one hard constraint that was not satisfied in the last instances. Instance JNS-CEJXXIII-2011-N got 25 best solutions. The gray cells show the best value found for that instance and red cells show values for unfeasible solutions.

Table 4.21: Best results, average and percentage ARD, with its score for each constraint, out of the 25 runs for the SA method where $t = 100000$, $\alpha = 0.5$, and $t = 1$ when the solution is feasible

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	270.76	0.654275
FA-EEF-2011-M	0	0	0	6	15	234	255	257.44	0.956863
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254	0
CM-CEDB-2010-N	0	0	0	10	0	288	298	298.2	0.0671141
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	320.56	0.489028
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	324.44	0.44582
JNS-CEDPII-2011-V	0	0	0	8	0	450	458	465.32	1.59825
CM-CECM-2011-N	0	0	0	44	33	612	689	703.96	2.17126
JNS-CEDPII-2011-M	0	0	0	12	3	468	483	491.16	1.68944
CL-CECL-2011-N-A	0	0	0	19	6	603	628	630.84	0.452229
MGA-CEVB-2011-V	0	0	0	16	3	540	559	564.96	1.06619
MGA-CEVB-2011-M	0	0	0	18	3	558	579	589	1.72712
CL-CEASD-2008-V-A	0	0	0	30	15	666	711	718.48	1.05204
CL-CEASD-2008-V-B	0	0	0	26	18	666	710	716.36	0.895775
MGA-CEDC-2011-V	0	0	0	34	21	684	739	749.04	1.35859
CL-CECL-2011-M-A	0	0	0	21	18	711	750	755.84	0.778667
CL-CECL-2011-M-B	0	0	0	22	9	711	742	750.4	1.13208
CM-CECM-2011-V	0	0	0	39	21	765	825	843.72	2.26909
CL-CECL-2011-V-A	0	0	0	28	0	747	775	787.28	1.58452
CM-CEUP-2008-V	0	0	0	68	27	918	1013	1031.6	1.83613
CM-CEUP-2011-M	0	0	0	72	48	945	1065	1085.88	1.96056
CM-CEUP-2011-V	0	0	0	41	33	891	965	979.28	1.47979
MGA-CEJXXIII-2010-V	0	0	0	57	18	873	948	975.2	2.8692
NE-CESVP-2011-V-A	0	0	0	43	24	972	1039	1054.96	1.53609
NE-CESVP-2011-V-B	0	0	0	39	30	972	1041	1057.96	1.6292
NE-CESVP-2011-V-C	0	0	0	39	39	954	1032	1050.76	1.81783
NE-CESVP-2011-M-A	0	0	0	48	30	1071	1149	1162.72	1.19408
NE-CESVP-2011-M-B	0	0	0	52	21	1071	1144	1158.8	1.29371
NE-CESVP-2011-M-C	0	0	0	55	27	1080	1162	1176.76	1.27022
NE-CESVP-2011-M-D	0	0	0	51	21	1089	1161	1177.72	1.44014
MGA-CEDC-2011-M	0	0	0	74	21	1008	1103	1113.96	0.993654
CM-CECM-2011-M	0	0	0	81	21	1188	1290	1310.4	1.5814
MGA-CEGV-2011-M	0	0	0	129	33	1800	1962	1991.76	1.51682
MGA-CEGV-2011-V	0	0	0	115	51	1971	2137	2178.36	1.93542

Table 4.22: Results for the 25 runs using the SAI cooling scheme and 2TQ perturbation where temperature $t = 1$ and cooling rate $\alpha = 0.5$. Gray cells represent the best solutions; red cells represent unfeasible solutions.

	Runs																								
Instance	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
CM-CEUP-2011-N	269	269	269	269	269	269	269	269	269	269	269	269	269	269	269	270	269	269	269	269	270	269	269	269	269
FA-EEF-2011-M	255	255	255	256	255	255	255	255	255	255	256	256	255	255	255	255	255	255	255	255	255	255	256	255	255
JNS-CEJXXIII-2011-N	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254	254
CM-CEDB-2010-N	298	299	298	298	298	298	298	299	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	298	299
JNS-CEJXXIII-2011-M	320	319	320	319	319	320	319	319	321	319	322	321	323	319	319	319	319	322	319	319	319	319	319	319	319
JNS-CEJXXIII-2011-V	325	324	323	324	323	324	323	324	324	323	325	323	323	323	325	323	323	323	323	323	323	323	325	324	325
JNS-CEDPII-2011-V	463	466	462	463	464	463	461	465	465	462	462	468	463	463	458	466	464	467	468	459	466	462	464	466	463
CM-CECM-2011-N	675	682	685	671	671	676	668	673	679	680	675	670	678	675	672	681	683	681	676	677	678	676	679	673	678
JNS-CEDPII-2011-M	488	491	487	493	486	486	483	489	494	484	487	481	496	486	488	486	487	484	486	485	488	487	484	488	489
CL-CECL-2011-N-A	628	631	633	628	629	632	628	630	628	634	630	632	633	636	629	632	634	630	633	633	633	630	635	631	629
MGA-CEVB-2011-V	566	559	567	559	564	557	562	563	560	564	562	560	557	556	567	561	558	559	559	559	563	561	563	557	560
MGA-CEVB-2011-M	599	586	582	585	608	583	585	584	589	581	585	582	579	576	587	593	581	590	585	583	593	581	588	587	591
CL-CEASD-2008-V-A	708	715	716	715	726	718	711	714	719	720	721	718	716	722	717	715	714	720	715	720	722	713	715	713	711
CL-CEASD-2008-V-B	718	715	713	709	718	708	727	714	710	713	716	719	709	713	715	721	715	716	723	717	718	717	719	715	712
MGA-CEDC-2011-V	733	741	738	741	741	743	731	737	739	743	746	735	733	734	735	734	739	739	744	729	736	739	737	744	741
CL-CECL-2011-M-A	758	763	755	753	754	761	753	756	757	754	759	752	762	757	758	754	759	760	753	762	763	756	757	760	755
CL-CECL-2011-M-B	756	746	753	755	755	747	757	749	756	749	753	754	751	759	757	751	755	748	744	750	754	749	759	752	746
CM-CECM-2011-V	836	828	824	832	827	840	826	826	822	823	833	818	818	818	829	819	822	826	823	824	826	820	832	824	834
CL-CECL-2011-V-A	791	786	795	784	788	798	791	780	791	797	809	793	792	795	790	791	793	786	799	787	797	791	792	790	790
CM-CEUP-2008-V	1007	1013	1010	1019	1035	1012	1013	1004	1016	999	1020	1015	1012	1012	1010	1008	1028	1012	1008	1003	1010	1010	1014	1022	1011
CM-CEUP-2011-M	1059	1057	1066	1053	1052	1059	1064	1048	1050	1050	1060	1043	1047	1050	1064	1069	1053	1062	1063	1060	1053	1060	1050	1049	1050
CM-CEUP-2011-V	956	957	964	959	961	968	962	963	957	961	953	954	962	962	965	971	958	969	961	969	961	956	953	961	964
MGA-CEJXXIII-2010-V	955	952	941	944	958	961	945	968	949	955	967	949	938	963	960	950	947	955	958	948	955	964	948	945	958
NE-CESVP-2011-V-A	1057	1071	1057	1045	1052	1048	1049	1049	1048	1057	1048	1059	1047	1051	1065	1040	1049	1071	1041	1050	1058	1053	1042	1057	1054
NE-CESVP-2011-V-B	1053	1063	1048	1050	1047	1062	1052	1047	1051	1060	1050	1054	1065	1059	1054	1055	1058	1059	1051	1051	1054	1044	1049	1048	1045
NE-CESVP-2011-V-C	1054	1036	1041	1045	1050	1041	1033	1048	1038	1053	1035	1053	1044	1063	1056	1045	1045	1043	1035	1045	1052	1051	1045	1039	1042
NE-CESVP-2011-M-A	1160	1160	1157	1163	1153	1169	1163	1162	1167	1157	1168	1155	1159	1171	1177	1162	1148	1164	1158	1156	1155	1174	1166	1174	1155
NE-CESVP-2011-M-B	1152	1147	1160	1150	1160	1161	1148	1166	1148	1166	1162	1157	1149	1139	1147	1150	1151	1161	1176	1157	1162	1149	1163	1143	1155
NE-CESVP-2011-M-C	1180	1170	1176	1172	1175	1165	1168	1162	1166	1163	1167	1184	1168	1184	1174	1177	1158	1178	1169	1177	1190	1168	1177	1174	1156
NE-CESVP-2011-M-D	1165	1161	1167	1175	1195	1169	1158	1187	1178	1180	1181	1158	1153	1157	1168	1167	1168	1182	1167	1164	1170	1166	1159	1176	1174
MGA-CEDC-2011-M	1094	1099	1089	1090	1086	1084	1092	1084	1103	1089	1085	1094	1087	1084	1080	1087	1086	1080	1095	1088	1090	1091	1104	1082	1088
CM-CECM-2011-M	1284	1289	1280	1283	1281	1273	1285	1281	1278	1281	1279	1273	1298	1265	1290	1276	1282	1280	1287	1264	1282	1272	1271	1281	1285
MGA-CEGV-2011-M	1932	1940	1939	1949	1946	1930	1929	1940	1928	1942	1915	1945	1936	1937	1941	1926	1947	1945	1968	1933	1946	1959	1951	1920	1944
MGA-CEGV-2011-V	2147	2115	2144	2122	2097	2123	2109	2108	2094	2097	2109	2108	2119	2108	2123	2133	2106	2116	2119	2128	102133	2130	2111	2107	2129

Table 4.23 shows the best results obtained out of the 25 runs for this method. It successfully produced feasible schedules for each instance. However, the third soft constraint is the hardest to avoid since it affects the score the most even if it has a high score out of the three soft constraints. The average is from all the 25 values obtained for each method. The percentage ARD shows how much the average differs from the original. The maximum value for all instances that produced feasible schedules is 1.82639% for the instance MGA-CEVB-2011-M, and for all instances including the ones that produced at least one unfeasible solution is 192.139% for the instance MGA-CEGV-2011-V.

4.11 Best Result Comparison

As is seen on Table 4.24, the method that gave the best results was method with the SA cooling scheme with $\alpha = 0.5$ and $t = 1$, and it gave 22 best results, this could be because the relaxed rule of SA, which is more complex than RR, and its cooling rate accept worse solutions in order to explore the solution space even more. However, the SA cooling scheme has two parameters that need to be tuned. The method that gave the worst results (i.e. it did not give a best solution) was method CC and method RR, but method CC is the worst since it could not achieve at least a feasible solution for some instances. However, none of these methods could improve over Saviniec et al.'s ILS-TQ implementation. Method SC has the most consistent schedules throughout all instances: it got the least percentage ARD eight times for all instances. It is important to note that this consistency is only relevant if the best answer is a feasible schedule: it means that that method got rid of all hard constraints, but they were closer to the best one. Gray cells represent the best solutions, cyan cells represent better % ARD than Saviniec et al.'s, bold numbers are the lowest percentage ARD of our methods obtained for that instance, and an asterisk (*) means that the solution is the same as Saviniec's (which is our lower bound). All methods in the table, except for method CC got better % ARD in some instances. This table takes approximately 148 days to generate on a single processor.

Table 4.23: Best results, average and percentage ARD, with its score for each constraint, out of the 25 runs for the SAI method combined with 2TQ where $t = 1$ and $\alpha = 0.5$

Instance	hc_3	hc_4	hc_5	sc_1	sc_2	sc_3	Total	Average	% ARD
CM-CEUP-2011-N	0	0	0	11	15	243	269	269.08	0.0297398
FA-EEF-2011-M	0	0	0	6	15	234	255	255.16	0.0627451
JNS-CEJXXIII-2011-N	0	0	0	8	3	243	254	254	0
CM-CEDB-2010-N	0	0	0	10	0	288	298	298.12	0.0402685
JNS-CEJXXIII-2011-M	0	0	0	10	3	306	319	319.68	0.213166
JNS-CEJXXIII-2011-V	0	0	0	11	6	306	323	323.64	0.198142
JNS-CEDPII-2011-V	0	0	0	8	0	450	458	463.72	1.24891
CM-CECM-2011-N	0	0	0	29	36	603	668	676.48	1.26946
JNS-CEDPII-2011-M	0	0	0	10	3	468	481	487.32	1.31393
CL-CECL-2011-N-A	0	0	0	19	6	603	628	631.24	0.515924
MGA-CEVB-2011-V	0	0	0	16	0	540	556	560.92	0.884892
MGA-CEVB-2011-M	0	0	0	15	3	558	576	586.52	1.82639
CL-CEASD-2008-V-A	0	0	0	27	15	666	708	716.56	1.20904
CL-CEASD-2008-V-B	0	0	0	24	18	666	708	715.6	1.07345
MGA-CEDC-2011-V	0	0	0	27	18	684	729	738.08	1.24554
CL-CECL-2011-M-A	0	0	0	26	15	711	752	757.24	0.696809
CL-CECL-2011-M-B	0	0	0	27	6	711	744	752.2	1.10215
CM-CECM-2011-V	0	0	0	32	21	765	818	826	0.977995
CL-CECL-2011-V-A	0	0	0	30	3	747	780	791.84	1.51795
CM-CEUP-2008-V	0	0	0	60	21	918	999	1012.92	1.39339
CM-CEUP-2011-M	0	0	0	56	42	945	1043	1055.64	1.21189
CM-CEUP-2011-V	0	0	0	38	24	891	953	961.08	0.847849
MGA-CEJXXIII-2010-V	0	0	0	50	15	873	938	953.32	1.63326
NE-CESVP-2011-V-A	0	0	0	47	21	972	1040	1052.72	1.22308
NE-CESVP-2011-V-B	0	0	0	39	33	972	1044	1053.16	0.877395
NE-CESVP-2011-V-C	0	0	0	40	39	954	1033	1045.28	1.18877
NE-CESVP-2011-M-A	0	0	0	47	21	1080	1148	1162.12	1.22997
NE-CESVP-2011-M-B	0	0	0	47	30	1062	1139	1155.16	1.41879
NE-CESVP-2011-M-C	0	0	0	49	27	1080	1156	1171.92	1.37716
NE-CESVP-2011-M-D	0	0	0	52	21	1080	1153	1169.8	1.45707
MGA-CEDC-2011-M	0	0	0	57	15	1008	1080	1089.24	0.855556
CM-CECM-2011-M	0	0	0	73	30	1161	1264	1280	1.26582
MGA-CEGV-2011-M	0	0	0	112	30	1773	1915	1939.52	1.28042
MGA-CEGV-2011-V	0	0	0	117	33	1944	2094	6117.4	192.139

Table 4.24: Best result and ARD comparison of all methods, including Saviniec et al.'s implementation. An asterisk (*) means that the value of the solution is the same as Saviniec et al's. Gray cells represent the best solutions, bold numbers are the lowest percentage ARD of our methods obtained for that instance. The last three methods correspond to the SAI cooling scheme where $t = 1$, $t = 100000$ and $t = 1$ respectively. Cyan cells represent better % ARD than Saviniec et al.'s.

	Saviniec's		Our proposal																	
															Simulated Annealing Inner cooling scheme					
Instance	ILS-TQ		ILS-TQ		2TQ		RR		SC		CC		S3R		$\alpha = 0.5$		$\alpha = 0.1$		2TQ $\alpha = 0.5$	
	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD	Best	% ARD
CM-CEUP-2011-N	269	0.18	*	0.09	*	**0.01**	270	0.28	*	**0.01**	275	113600	*	0.07	*	0.1	*	0.65	*	0.03
FA-EEF-2011-M	254	0.38	*	0.49	*	0.46	267	3.78	*	0.43	271	59634.3	255	0.08	*	0.52	255	0.96	255	**0.06**
JNS-CEJXXIII-2011-N	254	0	*	**0**	*	**0**	266	2.77	*	0.02	272	9.31	*	**0**	*	**0**	*	**0**	*	**0**
CM-CEDB-2010-N	298	0	*	0.05	*	0.07	328	3.9	*	0.38	329	8.15	*	**0**	*	**0**	*	0.07	*	0.04
JNS-CEJXXIII-2011-M	319	0.01	*	0.39	*	0.31	331	6.65	*	0.99	357	9.49	*	0.36	*	0.23	*	0.49	*	**0.21**
JNS-CEJXXIII-2011-V	323	0.09	*	0.3	*	0.19	341	3.91	*	0.38	366	8.01	*	0.41	*	**0.1**	*	0.45	*	0.2
JNS-CEDPII-2011-V	457	0.76	461	**0.77**	460	0.9	513	4.38	466	0.99	508	5.45	*	1.28	461	0.82	458	1.6	458	1.25
CM-CECM-2011-N	667	1.65	669	599.83	672	1.33	688	2.13	668	3.17	300700	74.1	671	**1.24**	668	1.62	689	2.17	668	1.27
JNS-CEDPII-2011-M	481	0.84	*	1.93	482	1.3	552	3.46	492	1.53	539	5.68	482	1.52	483	**0.73**	483	1.69	*	1.31
CL-CECL-2011-N-A	627	0.06	*	0.71	628	0.57	742	1.75	638	0.91	670	5.13	*	0.64	*	0.67	628	**0.45**	628	0.52
MGA-CEVB-2011-V	552	0.54	554	1.08	555	0.89	659	**0.78**	564	1.27	592	13920	555	1.08	555	0.87	559	1.07	556	0.88
MGA-CEVB-2011-M	571	1.37	576	1.91	577	1.91	688	1.91	593	**1.46**	628	3890.48	577	1.81	573	2.07	579	1.73	576	1.83
CL-CEASD-2008-V-A	699	0.57	704	1.31	708	1.01	844	1.76	729	1.14	767	2089.75	706	1.21	705	**0.91**	711	1.05	708	1.21
CL-CEASD-2008-V-B	699	0.82	700	1.76	708	**0.84**	837	2.82	729	1.34	769	523.85	701	1.89	706	1.07	710	0.9	708	1.07
MGA-CEDC-2011-V	728	0.47	734	2725.54	732	3279.55	816	1.32	750	**1.02**	110792	265.37	733	1092.41	730	1096.85	739	1.36	729	1.25
CL-CECL-2011-M-A	744	0.31	749	1.03	754	0.75	1013	2.5	765	3.42	829	3.4	753	0.8	751	0.97	750	0.78	752	**0.7**
CL-CECL-2011-M-B	736	0.44	742	1.21	745	**0.82**	1027	2	773	1.53	822	4.3	745	1.06	740	1.28	742	1.13	744	1.1
CM-CECM-2011-V	806	1.49	818	1.44	811	2.22	911	1.23	834	1.72	10887	2256.03	816	1.75	814	1.7	825	2.27	818	**0.98**
CL-CECL-2011-V-A	772	0.31	781	1.36	783	1.19	1090	3.78	813	**1.12**	853	4.74	777	1.96	778	1.25	775	1.58	780	1.52
CM-CEUP-2008-V	985	0.69	1000	1.75	1002	1.42	1124	3.71	1045	**1.26**	1120	22856.4	1004	1.32	*	2.29	1013	1.84	999	1.39
CM-CEUP-2011-M	1028	0.83	1039	386.31	1046	1148.03	1148	1.36	1084	2251.86	301121	137.88	1045	39.38	1036	**1.14**	1065	1.96	1043	1.21
CM-CEUP-2011-V	940	0.98	944	849.28	953	1.17	1084	**0.65**	980	1.03	201021	123.97	950	1.28	944	1.7	965	1.48	953	0.85
MGA-CEJXXIII-2010-V	923	1.39	941	1.46	935	2.21	1090	2.1	978	**0.95**	1017	13494.7	943	1.44	941	1.44	948	2.87	938	1.63
NE-CESVP-2011-V-A	1027	0.62	1035	1.83	1041	1.44	1280	1.52	1087	**0.9**	1145	3.29	1040	1.43	1033	1.35	1039	1.54	1040	1.22
NE-CESVP-2011-V-B	1028	0.86	1045	1	1041	1.61	1271	1.32	1073	1.88	1140	705.21	1043	1.23	1039	1.3	1041	1.63	1044	**0.88**
NE-CESVP-2011-V-C	1020	0.83	1033	1.41	1035	1.55	1230	1.96	1071	1.56	1158	347.44	1038	1.24	1029	1.56	1032	1.82	1033	**1.19**
NE-CESVP-2011-M-A	1130	0.66	1153	0.99	1158	0.87	1443	1.67	1204	0.86	1256	3.3	1157	1	1149	**0.74**	1149	1.19	1148	1.23
NE-CESVP-2011-M-B	1121	0.88	1151	**0.94**	1148	1.08	1432	1.18	1195	1.11	1241	3.66	1143	2.01	1140	1.28	1144	1.29	1139	1.42
NE-CESVP-2011-M-C	1141	0.8	1157	1.77	1167	1.16	1469	1.98	1211	**1.06**	1265	319.02	1161	1.71	1155	1.37	1162	1.27	1156	1.38
NE-CESVP-2011-M-D	1138	0.77	1163	1.39	1161	**1.35**	1398	1.92	1200	1.53	1263	3.54	1159	1.99	1153	1.52	1161	1.44	1153	1.46
MGA-CEDC-2011-M	1058	0.75	1079	**0.79**	1080	1	1242	1.04	1114	0.91	101150	342.48	1082	1.02	1073	1.08	1103	0.99	1080	0.86
CM-CECM-2011-M	1243	0.93	1274	628.82	1261	33.68	1497	**0.94**	1310	1.38	1383	14229.8	1272	1.29	1258	1.24	1290	1.58	1264	1.27
MGA-CEGV-2011-M	1865	0.83	1930	1.3	1926	1.75	2391	**0.93**	2004	1.04	2109	7547.7	1936	1.54	1913	1.12	1962	1.52	1915	1.28
MGA-CEGV-2011-V	2037	1.13	2110	191.32	2119	1.29	2485	**0.82**	2183	1.13	102256	248.8	2128	188.85	2081	1.07	2137	1.94	2094	192.14
Best count	34	26	13	4	8	5	0	5	7	**8**	0	0	7	3	22	7	6	2	11	7

64

Figure 4.1 compares our best method SAI with $\alpha = 0.5$ and $t = 1$ to our implementation of the original ILS-TQ method after running once for instances 33 and 34. The X axis represents the number of seconds and the Y axis represents the scores of each schedule. Our best method found a better solution in less seconds.

4.12 Schedule Improvement

In this section, schedules of our methods for the first instance (i.e. CM-CEUP-2011-N) are shown here, and how they evolve at the $10th$, $50th$, $100th$ and final iterations. Each schedule is represented by the teachers' ids, and the requirements' ids in order to see better the constraint violations. This run is unrelated to the 25 runs previously done.

Table 4.25 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for our ILS-TQ implementation. It already found a feasible solution by the $10th$ iteration and slowly improved until the final iteration, it improved seven points from 276 to 269. It found the lower bound by the final iteration, but got stuck from the $50th$ iteration to the $100th$ iteration.

Table 4.26 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method 2TQ. It found a feasible solution by the $100th$ iteration and slowly improved until the end. It improved four points from 273 to 269; it found the lower bound.

Table 4.27 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method RR. It found a feasible solution by the tenth iteration, but it did not get the lower bound by the final iteration.

Table 4.28 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method SC. It found a feasible solution by the $50th$ iteration and improved one point until the end, this method could also find the schedule with score of 269 which is the lower bound.

Table 4.29 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method CC. It found no feasible solution, and it never improved. Furthermore, it did not find a better schedule than the one found by the $10th$ iteration because all the schedules are the same. However, there are no teacher clashes, and there are five hard constraints that were not satisfied.

Table 4.30 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method S3R. It found a feasible solution by the $50th$ iteration, and it slowly improved until the final iteration, it improved three points from 273 to 270.

Table 4.31 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method SAI when $\alpha = 0.5$ and $t = 1$. It found a feasible solution by the $10th$ iteration, and it slowly improved until the final iteration, it improved four points from 274 to 270. This method did not get the lower bound.

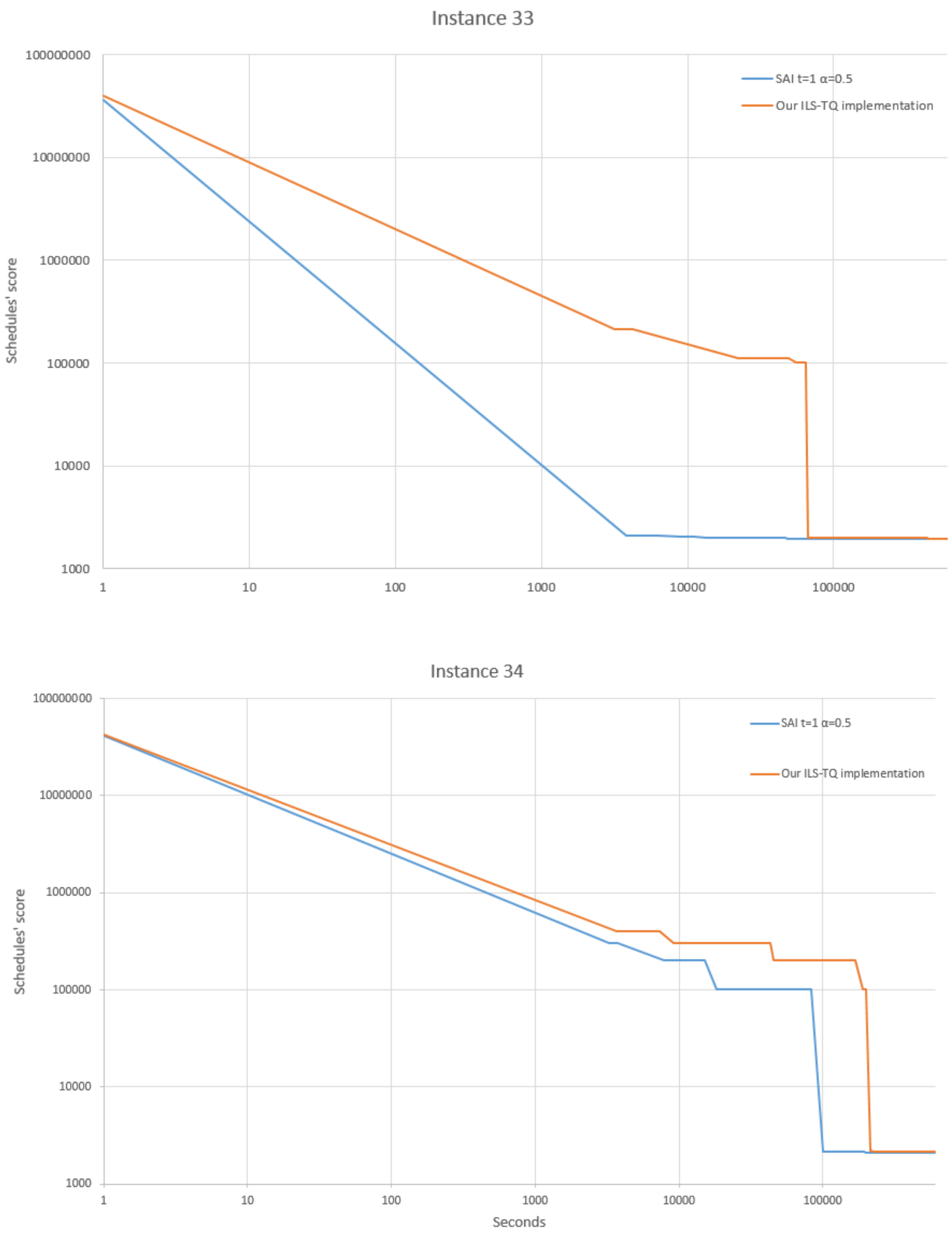


Figure 4.1: Our best method (blue line) compared to our ILS-TQ implementation (orange line) after running once for instances 33 and 34. Our best method not only gave a better solution, but it also found a better solution in less time compared to our ILS-TQ implementation

Table 4.25: Schedules generated by our ILS-TQ implementation. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

ILS-TQ - 10th iteration - Score: 276

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	8	13	3	13	4	1	4	9	4	14	0	3	14	11	11
c_1	9	13	7	7	8	7	7	5	5	12	4	3	8	4	13	9	6	6	12	4	3	0	0	14	14
c_2	7	7	6	13	13	1	5	7	7	5	3	4	4	8	8	6	1	4	9	9	14	14	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	8	13	3	13	4	1	4	9	4	14	0	3	14	11	11
c_1	24	28	22	22	23	22	22	20	20	27	19	18	23	19	28	24	21	21	27	19	18	15	15	29	29
c_2	37	37	36	43	43	31	35	37	37	35	33	34	34	38	38	36	31	34	39	39	44	44	33	30	30

ILS-TQ - 50th iteration - Score: 274

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	3	13	8	13	4	9	1	4	4	14	0	3	14	11	11
c_1	9	13	7	7	8	7	5	5	7	12	8	3	4	4	13	6	6	9	12	4	3	0	0	14	14
c_2	7	7	6	13	13	1	7	7	5	5	4	4	3	8	8	1	4	6	9	9	14	14	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	3	13	8	13	4	9	1	4	4	14	0	3	14	11	11
c_1	24	28	22	22	23	22	20	20	22	27	23	18	19	19	28	21	21	24	27	19	18	15	15	29	29
c_2	37	37	36	43	43	31	37	37	35	35	34	34	33	38	38	31	34	36	39	39	44	44	33	30	30

ILS-TQ - 100th iteration - Score: 274

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	3	13	8	13	4	9	1	4	4	14	0	3	14	11	11
c_1	9	13	7	7	8	7	7	5	5	12	8	3	4	4	13	6	6	9	12	4	3	0	0	14	14
c_2	7	7	6	13	13	1	5	7	7	5	4	4	3	8	8	1	4	6	9	9	14	14	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	3	13	8	13	4	9	1	4	4	14	0	3	14	11	11
c_1	24	28	22	22	23	22	22	20	20	27	23	18	19	19	28	21	21	24	27	19	18	15	15	29	29
c_2	37	37	36	43	43	31	35	37	37	35	34	34	33	38	38	31	34	36	39	39	44	44	33	30	30

ILS-TQ - Final iteration - Score: 269

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	0	2	2	3	13	3	4	4	4	6	6	9	9	0	14	14	11	11
c_1	7	7	6	13	13	7	5	5	7	12	6	4	4	8	8	9	9	4	12	14	14	3	3	0	0
c_2	6	6	7	7	8	9	7	7	5	5	4	3	8	13	13	1	1	9	4	4	3	0	0	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	0	2	2	3	13	3	4	4	4	6	6	9	9	0	14	14	11	11
c_1	22	22	21	28	28	22	20	20	22	27	21	19	19	23	23	24	24	19	27	29	29	18	18	15	15
c_2	36	36	37	37	38	39	37	37	35	35	34	33	38	43	43	31	31	39	34	34	33	30	30	44	44

Table 4.26: Schedules generated by the method 2TQ. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

2TQ - 10th iteration - Score: 200272

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	0	0	9	2	2	8	13	3	4	4	1	1	6	9	4	14	14	3	11	11
c_1	9	7	7	13	13	7	5	5	12	12	3	3	4	8	8	6	6	4	4	9	0	0	14	14	7
c_2	7	6	6	7	8	1	7	7	5	5	4	4	8	13	13	4	9	9	1	14	3	3	0	0	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	0	0	9	2	2	8	13	3	4	4	1	1	6	9	4	14	14	3	11	11
c_1	24	22	22	28	28	22	20	20	27	27	18	18	19	23	23	21	21	19	19	24	15	15	29	29	22
c_2	37	36	36	37	38	31	37	37	35	35	34	34	38	43	43	34	39	39	31	44	33	33	30	30	44

2TQ - 50th iteration - Score: 100273

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	0	0	2	2	8	3	4	4	13	1	6	6	9	4	14	14	3	11	11
c_1	7	7	6	13	13	7	7	5	5	12	4	4	3	8	8	6	4	9	12	9	3	0	0	14	14
c_2	6	6	7	7	8	5	1	7	7	5	3	13	8	13	4	9	9	4	4	14	1	3	14	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	0	0	2	2	8	3	4	4	13	1	6	6	9	4	14	14	3	11	11
c_1	22	22	21	28	28	22	22	20	20	27	19	19	18	23	23	21	19	24	27	24	18	15	15	29	29
c_2	36	36	37	37	38	35	31	37	37	35	33	43	38	43	34	39	39	34	34	44	31	33	44	30	30

2TQ - 100th iteration - Score: 273

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	4	3	13	13	6	9	9	4	14	0	3	14	11	11
c_1	7	7	6	13	13	7	7	5	5	12	6	3	8	8	4	9	4	4	12	9	14	14	3	0	0
c_2	6	13	7	7	8	9	5	7	7	5	3	13	4	4	8	1	1	6	9	4	3	0	0	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	4	3	13	13	6	9	9	4	14	0	3	14	11	11
c_1	22	22	21	28	28	22	22	20	20	27	21	18	23	23	19	24	19	19	27	24	29	29	18	15	15
c_2	36	43	37	37	38	39	35	37	37	35	33	43	34	34	38	31	31	36	39	34	33	30	30	44	44

2TQ - Final iteration - Score: 269

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	4	6	6	4	14	0	0	14	11	11
c_1	7	7	6	13	13	5	5	7	7	12	4	4	3	8	8	6	9	9	12	4	14	14	3	0	0
c_2	6	6	7	7	8	7	7	5	5	0	8	13	4	4	13	1	1	4	9	9	3	3	0	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	4	6	6	4	14	0	0	14	11	11
c_1	22	22	21	28	28	20	20	22	22	27	19	19	18	23	23	21	24	24	27	19	29	29	18	15	15
c_2	36	36	37	37	38	37	37	35	35	30	38	43	34	34	43	31	31	34	39	39	33	33	30	44	44

Table 4.27: Schedules generated by the method RR. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

RR - 10th iteration - Score: 285

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	6	10	10	1	9	0	2	2	4	13	3	4	8	1	6	9	4	14	14	3	0	11	11
c_1	6	6	7	7	13	7	7	5	5	12	3	4	8	8	13	9	9	4	12	4	0	14	3	14	0
c_2	7	7	8	13	8	5	1	7	7	5	6	3	4	13	4	4	1	6	9	9	3	0	14	0	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	6	10	10	1	9	0	2	2	4	13	3	4	8	1	6	9	4	14	14	3	0	11	11
c_1	21	21	22	22	28	22	22	20	20	27	18	19	23	23	28	24	24	19	27	19	15	29	18	29	15
c_2	37	37	38	43	38	35	31	37	37	35	36	33	34	43	34	34	31	36	39	39	33	30	44	30	44

RR - 50th iteration - Score: 279

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	1	1	0	2	2	3	13	4	4	8	6	9	9	4	14	0	3	14	11	11
c_1	7	6	6	7	13	7	7	5	5	12	4	3	8	8	13	9	4	4	12	9	14	0	3	14	0
c_2	8	7	7	13	8	9	5	7	7	5	6	4	3	13	4	1	1	6	9	4	3	14	0	0	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	1	1	0	2	2	3	13	4	4	8	6	9	9	4	14	0	3	14	11	11
c_1	22	21	21	22	28	22	22	20	20	27	19	18	23	23	28	24	19	19	27	24	29	15	18	29	15
c_2	38	37	37	43	38	39	35	37	37	35	36	34	33	43	34	31	31	36	39	34	33	44	30	30	44

RR - 100th iteration - Score: 279

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	1	1	0	2	2	3	13	4	4	8	6	9	9	4	14	0	3	14	11	11
c_1	7	6	6	7	13	7	7	5	5	12	4	3	8	8	13	9	4	4	12	9	14	0	3	14	0
c_2	8	7	7	13	8	9	5	7	7	5	6	4	3	13	4	1	1	6	9	4	3	14	0	0	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	1	1	0	2	2	3	13	4	4	8	6	9	9	4	14	0	3	14	11	11
c_1	22	21	21	22	28	22	22	20	20	27	19	18	23	23	28	24	19	19	27	24	29	15	18	29	15
c_2	38	37	37	43	38	39	35	37	37	35	36	34	33	43	34	31	31	36	39	34	33	44	30	30	44

RR - Final iteration - Score: 270

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	14	0	0	11	11
c_1	7	6	6	7	8	7	7	5	5	12	4	3	8	13	13	4	9	9	12	4	3	14	14	0	0
c_2	6	7	7	13	13	5	5	7	7	0	6	4	4	8	8	1	1	4	9	9	0	3	3	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	14	0	0	11	11
c_1	22	21	21	22	23	22	22	20	20	27	19	18	23	28	28	19	24	24	27	19	18	29	29	15	15
c_2	36	37	37	43	43	35	35	37	37	30	36	34	34	38	38	31	31	34	39	39	30	33	33	44	44

Table 4.28: Schedules generated by the method SC. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

SC - 10th iteration - Score: 100272

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	9	1	0	2	2	8	13	3	4	4	1	6	6	14	4	0	3	14	11	11
c_1	6	6	7	7	8	7	5	5	12	12	4	3	8	13	13	4	9	9	4	7	14	14	3	0	0
c_2	7	7	6	13	13	1	7	7	5	5	3	4	4	8	8	6	1	4	9	9	3	0	0	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	9	1	0	2	2	8	13	3	4	4	1	6	6	14	4	0	3	14	11	11
c_1	21	21	22	22	23	22	20	20	27	27	19	18	23	28	28	19	24	24	19	22	29	29	18	15	15
c_2	37	37	36	43	43	31	37	37	35	35	33	34	34	38	38	36	31	34	39	39	33	30	30	44	44

SC - 50th iteration - Score: 270

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	0	0	14	11	11
c_1	6	6	7	7	8	7	5	5	7	12	4	3	8	13	13	4	9	9	12	4	14	14	3	0	0
c_2	7	7	6	13	13	0	7	7	5	5	6	4	4	8	8	1	1	4	9	9	3	3	0	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	0	0	14	11	11
c_1	21	21	22	22	23	22	20	20	22	27	19	18	23	28	28	19	24	24	27	19	29	29	18	15	15
c_2	37	37	36	43	43	30	37	37	35	35	36	34	34	38	38	31	31	34	39	39	33	33	30	44	44

SC - 100th iteration - Score: 269

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	14	0	0	11	11
c_1	6	6	7	7	8	7	7	5	5	12	4	3	8	13	13	4	9	9	12	4	3	14	14	0	0
c_2	7	7	6	13	13	5	5	7	7	0	6	4	4	8	8	1	1	4	9	9	0	3	3	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	14	0	0	11	11
c_1	21	21	22	22	23	22	22	20	20	27	19	18	23	28	28	19	24	24	27	19	18	29	29	15	15
c_2	37	37	36	43	43	35	35	37	37	30	36	34	34	38	38	31	31	34	39	39	30	33	33	44	44

SC - Final iteration - Score: 269

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	0	0	14	11	11
c_1	7	7	6	13	13	5	5	7	7	12	6	3	4	8	8	4	9	9	12	4	14	14	3	0	0
c_2	6	6	7	7	8	7	7	5	5	0	4	4	8	13	13	1	1	4	9	9	3	3	0	14	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	9	2	2	3	13	3	4	4	9	6	6	4	14	0	0	14	11	11
c_1	22	22	21	28	28	20	20	22	22	27	21	18	19	23	23	19	24	24	27	19	29	29	18	15	15
c_2	36	36	37	37	38	37	37	35	35	30	34	34	38	43	43	31	31	34	39	39	33	33	30	44	44

Table 4.29: Schedules generated by the method CC. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

CC - 10th iteration - Score: 500280																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	6	6	7	7	8	5	5	7	7	12	8	4	4	13	13	9	9	4	12	14	3	0	14	3	0
c_2	7	7	6	13	13	1	1	5	5	7	4	3	3	8	8	6	4	9	9	4	14	14	0	0	7
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	21	21	22	22	23	20	20	22	22	27	23	19	19	28	28	24	24	19	27	29	18	15	29	18	15
c_2	37	37	36	43	43	31	31	35	35	37	34	33	33	38	38	36	34	39	39	34	44	44	30	30	37

CC - 50th iteration - Score: 500280																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	6	6	7	7	8	5	5	7	7	12	8	4	4	13	13	9	9	4	12	14	3	0	14	3	0
c_2	7	7	6	13	13	1	1	5	5	7	4	3	3	8	8	6	4	9	9	4	14	14	0	0	7
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	21	21	22	22	23	20	20	22	22	27	23	19	19	28	28	24	24	19	27	29	18	15	29	18	15
c_2	37	37	36	43	43	31	31	35	35	37	34	33	33	38	38	36	34	39	39	34	44	44	30	30	37

CC - 100th iteration - Score: 500280																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	6	6	7	7	8	5	5	7	7	12	8	4	4	13	13	9	9	4	12	14	3	0	14	3	0
c_2	7	7	6	13	13	1	1	5	5	7	4	3	3	8	8	6	4	9	9	4	14	14	0	0	7
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	21	21	22	22	23	20	20	22	22	27	23	19	19	28	28	24	24	19	27	29	18	15	29	18	15
c_2	37	37	36	43	43	31	31	35	35	37	34	33	33	38	38	36	34	39	39	34	44	44	30	30	37

CC - Final iteration - Score: 500280																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	6	6	7	7	8	5	5	7	7	12	8	4	4	13	13	9	9	4	12	14	3	0	14	3	0
c_2	7	7	6	13	13	1	1	5	5	7	4	3	3	8	8	6	4	9	9	4	14	14	0	0	7
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	11	11	0	2	2	6	13	8	4	4	1	1	6	4	9	0	3	3	14	14
c_1	21	21	22	22	23	20	20	22	22	27	23	19	19	28	28	24	24	19	27	29	18	15	29	18	15
c_2	37	37	36	43	43	31	31	35	35	37	34	33	33	38	38	36	34	39	39	34	44	44	30	30	37

Table 4.30: Schedules generated by the method S3R. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

S3R - 10th iteration - Score: 100270

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	0	9	9	2	2	8	3	4	13	13	1	1	4	4	14	14	3	0	11	11
c_1	7	13	6	7	8	7	7	5	5	12	3	13	8	4	4	6	9	9	12	4	0	0	3	14	14
c_2	5	7	7	13	13	1	1	7	7	5	4	4	3	8	8	4	6	6	9	9	3	14	14	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	0	9	9	2	2	8	3	4	13	13	1	1	4	4	14	14	3	0	11	11
c_1	22	28	21	22	23	22	22	20	20	27	18	28	23	19	19	21	24	24	27	19	15	15	18	29	29
c_2	35	37	37	43	43	31	31	37	37	35	34	34	33	38	38	34	36	36	39	39	33	44	44	30	30

S3R - 50th iteration - Score: 273

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	0	1	9	2	2	4	3	8	13	13	1	9	4	4	14	0	3	14	11	11
c_1	9	13	7	7	8	7	7	5	5	12	8	13	3	4	4	6	6	9	12	4	3	0	0	14	14
c_2	7	7	6	13	13	1	5	7	7	5	3	4	4	8	8	4	1	6	9	9	14	14	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	0	1	9	2	2	4	3	8	13	13	1	9	4	4	14	0	3	14	11	11
c_1	24	28	22	22	23	22	22	20	20	27	23	28	18	19	19	21	21	24	27	19	18	15	15	29	29
c_2	37	37	36	43	43	31	35	37	37	35	33	34	34	38	38	34	31	36	39	39	44	44	33	30	30

S3R - 100th iteration - Score: 273

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	4	3	8	13	13	9	1	4	4	14	0	3	14	11	11
c_1	8	13	7	7	8	7	7	5	5	12	6	13	3	4	4	6	9	9	12	4	3	0	0	14	14
c_2	7	7	6	13	13	1	5	7	7	5	3	4	4	8	8	1	4	6	9	9	14	14	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	9	1	0	2	2	4	3	8	13	13	9	1	4	4	14	0	3	14	11	11
c_1	23	28	22	22	23	22	22	20	20	27	21	28	18	19	19	21	24	24	27	19	18	15	15	29	29
c_2	37	37	36	43	43	31	35	37	37	35	33	34	34	38	38	31	34	36	39	39	44	44	33	30	30

S3R - Final iteration - Score: 270

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	3	3	13	13	6	4	4	9	9	0	14	14	11	11
c_1	7	7	6	13	13	7	5	5	7	12	3	4	4	8	8	9	9	6	12	4	3	0	0	14	14
c_2	6	13	7	7	8	9	7	7	5	5	6	13	8	4	4	1	1	9	4	14	14	3	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	3	3	13	13	6	4	4	9	9	0	14	14	11	11
c_1	22	22	21	28	28	22	20	20	22	27	18	19	19	23	23	24	24	21	27	19	18	15	15	29	29
c_2	36	43	37	37	38	39	37	37	35	35	36	43	38	34	34	31	31	39	34	44	44	33	33	30	30

Table 4.31: Schedules generated by the method SAI when $\alpha = 0.5$ and $t = 1$. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

SAI $\alpha = 0.5$ and $t = 1$ - 10th iteration - Score: 274

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	1	1	0	2	2	3	3	8	13	4	9	9	6	4	4	0	14	14	11	11
c_1	7	7	6	13	13	7	7	5	5	12	4	4	3	8	8	4	6	9	12	9	14	0	3	14	0
c_2	8	6	7	7	8	9	5	7	7	5	6	13	4	4	13	1	1	4	9	14	3	3	0	0	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	13	8	10	10	1	1	0	2	2	3	3	8	13	4	9	9	6	4	4	0	14	14	11	11
c_1	22	22	21	28	28	22	22	20	20	27	19	19	18	23	23	19	21	24	27	24	29	15	18	29	15
c_2	38	36	37	37	38	39	35	37	37	35	36	43	34	34	43	31	31	34	39	44	33	33	30	30	44

SAI $\alpha = 0.5$ and $t = 1$ - 50th iteration - Score: 271

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	3	3	4	13	13	9	9	6	4	4	0	14	14	11	11
c_1	7	7	6	13	13	7	5	5	7	12	4	4	3	8	8	6	4	9	12	9	3	0	0	14	14
c_2	6	13	7	7	8	9	7	7	5	5	6	13	8	4	4	1	1	4	9	14	14	3	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	3	3	4	13	13	9	9	6	4	4	0	14	14	11	11
c_1	22	22	21	28	28	22	20	20	22	27	19	19	18	23	23	21	19	24	27	24	18	15	15	29	29
c_2	36	43	37	37	38	39	37	37	35	35	36	43	38	34	34	31	31	34	39	44	44	33	33	30	30

SAI $\alpha = 0.5$ and $t = 1$ - 100th iteration - Score: 270

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	3	3	13	13	6	4	4	9	9	0	14	14	11	11
c_1	7	7	6	13	13	7	5	5	7	12	3	4	4	8	8	9	9	6	12	4	3	0	0	14	14
c_2	6	13	7	7	8	9	7	7	5	5	6	13	8	4	4	1	1	9	4	14	14	3	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	3	3	13	13	6	4	4	9	9	0	14	14	11	11
c_1	22	22	21	28	28	22	20	20	22	27	18	19	19	23	23	24	24	21	27	19	18	15	15	29	29
c_2	36	43	37	37	38	39	37	37	35	35	36	43	38	34	34	31	31	39	34	44	44	33	33	30	30

SAI $\alpha = 0.5$ and $t = 1$ - Final iteration - Score: 270

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	0	2	2	3	13	3	4	4	4	6	6	9	9	0	14	14	11	11
c_1	7	7	6	13	13	7	7	5	5	12	6	3	4	8	8	9	9	4	12	4	3	0	0	14	14
c_2	6	6	7	7	8	9	5	7	7	5	4	4	8	13	13	1	1	9	4	14	14	3	3	0	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	1	0	2	2	3	13	3	4	4	4	6	6	9	9	0	14	14	11	11
c_1	22	22	21	28	28	22	22	20	20	27	21	18	19	23	23	24	24	19	27	19	18	15	15	29	29
c_2	36	36	37	37	38	39	35	37	37	35	34	34	38	43	43	31	31	39	34	44	44	33	33	30	30

Table 4.32 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the method SAI when $\alpha = 0.5$ and $t = 100000$. It found a feasible solution by the $50th$ iteration, and it slowly improved until the final iteration, it improved 23 points from 294 to 271. The final schedule is not the lower bound, since there is a better schedule with score 269.

Table 4.33 shows the schedules generated at the $10th$, $50th$, $100th$ and final iterations for the hybrid method SAI with 2TQ when $\alpha = 0.5$ and $t = 1$. It found a feasible solution by the $10th$ iteration, and it slowly improved until the final iteration, it improved four points from 273 to 269 by the $100th$ iteration.

It is interesting to note that one method can get the lower bound (i.e. best result for that instance) for one instance even if in the first iterations it finds worse results than other methods. For example, the method 2TQ found the lower bound even if by the $100th$ iteration it did not get a feasible schedule, unlike methods CC, S3R, the SAI with $\alpha = 0.5, t = 1$ and the SAI with $\alpha = 0.5, t = 100000$ that did not get the lower bound. The method CC performed the worst and did not find a feasible solution; in fact, it did not improve its schedule. None of the methods gave schedules that violated the hard constraint that had to do with teacher clashes (hard constraint hc_3), at least by the $10th$ iteration. However, the hard constraint hc_4 was present in unfeasible schedules found by all methods. Schedules may differ even if their scores are the same. Finally, these methods try to satisfy all constraints even if their schedules are unfeasible.

Table 4.32: Schedules generated by the method SAI when $\alpha = 0.5$ and $t = 100000$. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

SAI $\alpha = 0.5$ and $t = 100000$ - 10th iteration - Score: 100301

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	9	0	2	2	6	3	4	4	13	1	6	4	14	9	0	3	14	11	11
c_1	6	6	7	7	13	7	5	7	5	12	4	4	8	13	8	9	4	9	12	14	3	0	3	0	14
c_2	7	7	6	13	8	9	1	5	7	5	3	13	3	8	4	4	1	6	9	4	7	14	0	14	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	13	8	10	10	1	9	0	2	2	6	3	4	4	13	1	6	4	14	9	0	3	14	11	11
c_1	21	21	22	22	28	22	20	22	20	27	19	19	23	28	23	24	19	24	27	29	18	15	18	15	29
c_2	37	37	36	43	38	39	31	35	37	35	33	43	33	38	34	34	31	36	39	34	37	44	30	44	30

SAI $\alpha = 0.5$ and $t = 100000$ - 50th iteration - Score: 294

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	0	9	2	2	3	13	4	4	13	1	6	9	4	14	14	0	3	11	11
c_1	7	13	6	7	8	7	7	5	5	12	4	3	8	13	4	9	4	6	12	9	0	3	14	14	0
c_2	9	7	7	13	13	5	1	7	7	5	6	4	3	8	8	6	1	4	9	4	3	14	0	0	14

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	0	9	2	2	3	13	4	4	13	1	6	9	4	14	14	0	3	11	11
c_1	22	28	21	22	23	22	22	20	20	27	19	18	23	28	19	24	19	21	27	24	15	18	29	29	15
c_2	39	37	37	43	43	35	31	37	37	35	36	34	33	38	38	36	31	34	39	34	33	44	30	30	44

SAI $\alpha = 0.5$ and $t = 100000$ - 100th iteration - Score: 275

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	9	1	0	2	2	4	3	3	13	13	4	1	6	4	9	14	14	0	11	11
c_1	6	13	7	7	8	7	7	5	5	12	6	13	8	4	4	9	9	4	12	14	0	3	3	0	14
c_2	7	7	6	13	13	1	5	7	7	5	3	4	4	8	8	1	6	9	9	4	3	0	14	14	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	9	1	0	2	2	4	3	3	13	13	4	1	6	4	9	14	14	0	11	11
c_1	21	28	22	22	23	22	22	20	20	27	21	28	23	19	19	24	24	19	27	29	15	18	18	15	29
c_2	37	37	36	43	43	31	35	37	37	35	33	34	34	38	38	31	36	39	39	34	33	30	44	44	30

SAI $\alpha = 0.5$ and $t = 100000$ - Final iteration - Score: 271

Teachers

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	4	6	6	4	14	0	0	14	11	11
c_1	7	7	6	13	13	7	7	5	5	12	4	4	3	8	8	6	9	9	12	4	3	14	0	0	14
c_2	6	6	7	7	8	5	5	7	7	0	8	13	4	4	13	1	1	4	9	9	14	3	3	14	0

Requirements

	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	4	6	6	4	14	0	0	14	11	11
c_1	22	22	21	28	28	22	22	20	20	27	19	19	18	23	23	21	24	24	27	19	18	29	15	15	29
c_2	36	36	37	37	38	35	35	37	37	30	38	43	34	34	43	31	31	34	39	39	44	33	33	44	30

Table 4.33: Schedules generated by the hybrid method SAI with 2TQ operator when $\alpha = 0.5$ and $t = 1$. There is one schedule per iteration, the upper one shows the teachers' ids and the lower one shows the requirements' ids

SAI with 2TQ $\alpha = 0.5$ and $t = 1$ - 10th iteration - Score: 100276																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	1	1	0	2	2	8	4	4	13	13	4	9	9	14	14	3	3	0	11	11
c_1	7	7	6	13	13	7	5	5	12	12	3	3	8	8	4	6	4	4	9	9	0	0	14	14	7
c_2	9	13	7	7	8	9	7	7	5	5	6	13	3	4	8	1	1	6	4	4	14	14	3	0	0
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	6	6	8	10	10	1	1	0	2	2	8	4	4	13	13	4	9	9	14	14	3	3	0	11	11
c_1	22	22	21	28	28	22	20	20	27	27	18	18	23	23	19	21	19	19	24	24	15	15	29	29	22
c_2	39	43	37	37	38	39	37	37	35	35	36	43	33	34	38	31	31	36	34	34	44	44	33	30	30

SAI with 2TQ $\alpha = 0.5$ and $t = 1$ - 50th iteration - Score: 273																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	4	3	13	13	6	4	9	9	14	14	3	0	11	11
c_1	7	7	6	13	13	7	5	5	7	12	6	3	4	8	8	9	9	4	12	4	3	14	14	0	0
c_2	6	13	7	7	8	9	7	7	5	5	3	13	8	4	4	1	1	6	4	9	0	0	3	14	14
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	8	6	8	10	10	1	1	0	2	2	4	4	3	13	13	6	4	9	9	14	14	3	0	11	11
c_1	22	22	21	28	28	22	20	20	22	27	21	18	19	23	23	24	24	19	27	19	18	29	29	15	15
c_2	36	43	37	37	38	39	37	37	35	35	33	43	38	34	34	31	31	36	34	39	30	30	33	44	44

SAI with 2TQ $\alpha = 0.5$ and $t = 1$ - 100th iteration - Score: 269																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	6	6	4	4	14	14	0	0	11	11
c_1	7	7	6	13	13	7	7	5	5	12	4	4	3	8	8	4	9	6	12	9	3	14	14	0	0
c_2	6	6	7	7	8	5	5	7	7	0	8	13	4	4	13	1	1	9	9	4	0	3	3	14	14
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	6	6	4	4	14	14	0	0	11	11
c_1	22	22	21	28	28	22	22	20	20	27	19	19	18	23	23	19	24	21	27	24	18	29	29	15	15
c_2	36	36	37	37	38	35	35	37	37	30	38	43	34	34	43	31	31	39	39	34	30	33	33	44	44

SAI with 2TQ $\alpha = 0.5$ and $t = 1$ - Final iteration - Score: 269																									
Teachers																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	4	6	6	4	14	14	0	0	11	11
c_1	7	7	6	13	13	5	5	7	7	12	4	4	3	8	8	6	9	9	12	4	3	14	14	0	0
c_2	6	6	7	7	8	7	7	5	5	0	8	13	4	4	13	1	1	4	9	9	0	3	3	14	14
Requirements																									
	Monday					Tuesday					Wednesday					Thursday					Friday				
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
c_0	9	13	8	10	10	1	1	9	2	2	3	3	8	13	4	4	6	6	4	14	14	0	0	11	11
c_1	22	22	21	28	28	20	20	22	22	27	19	19	18	23	23	21	24	24	27	19	18	29	29	15	15
c_2	36	36	37	37	38	37	37	35	35	30	38	43	34	34	43	31	31	34	39	39	30	33	33	44	44

Chapter 5

Concluding Remarks

In this thesis, the High school Timetabling Problem (HSTTP) was addressed. This problem consists in assigning subjects to timeslots while satisfying constraints, which are hard, and soft that represent feasible, and quality solutions respectively.

In Chapter 2, all techniques obtained feasible schedules, but not all techniques could be compared since the instances were different. How these techniques perturb their solutions also depends on how the schedule is represented, the constraints and also whether the method has a population of solutions or not. In our case, we had to adapt those operators to the problem we are trying to solve. ILS-TQ method by Saviniec and Constantino (2017) was chosen because it improved over GOAL's results, and it might show further improvement if its perturbation operator is improved.

In Chapter 3, we propose some modifications to the original ILS-TQ technique in the perturbation operator and relaxed rules. There were some modifications applied to how we encoded the inputs, and teachers' unavailabilities, we also explained the data structures we used to calculate each constraint. However, some of these data structures made our implementation just for these instances (i.e. instances that does not have the same teachers for the same class, and instances that has no more than 5 days or periods per day). We implemented the original ILS-TQ technique and modified it from there. Finally, the HSTTP instance creator makes it easier to create instances for our methods to try, but there is room for improvement.

In Chapter 4, the method that gave better results was our implementation of the SA cooling scheme inside the local search with $\alpha = 0.5$, $t = 1$ which generated 22 best schedules compared to our other methods since its complex relaxed rule accepts worse solutions to explore the solution space even more; however, it needs parameter tuning. The worst method was the Change Random Column (CC) one which obtained 0 best schedules, it only got one feasible solution in the first instance, and could not generate feasible schedules for all instances in 25 runs, unlike the rest that did; this method gave the worst results because it does not perturb the solution since the current solution and the best solution are always the same, so this method only locally searches once, but it gives more information about which instances need only one local search to find a feasible solution. Method RR also found no best solutions, but it always found feasible solutions for all runs. However, none of our methods improved over Saviniec et al.'s implementation (but our ILS-TQ implementation was improved); at most, the 2TQ, the SC and the best method generated schedules with the same results as Saviniec's for the first six instances. However, all of our methods, except for CC got better % ARD than Saviniec's for some instances. The method that had the most number of lowest percentage ARD is method SC which means that it gave the most stable schedules (i.e. schedules with similar scores). Schedules having similar scores does not mean that they are similar, even if they have the same score. Moreover, the fourth hard constraint is the most difficult to avoid, and the third hard constraint is the easiest one, even the CC method had no trouble removing all its violations. It is important to note that the S3R method successfully got a feasible solution by the $50th$ iteration, but got a worse result than the 2TQ method even if the latter did not get a feasible solution by the $100th$ iteration. Even if the schedules are not feasible, the methods still try to satisfy as many constraints as possible. Finally, consistent solutions are good in a method, but does not mean much if it does not get feasible schedules.

In Chapter 4, it can also be seen that what worked for some authors does not mean it works for us: we implemented or adapted their operators and relaxed rules to the original method and not all of them performed better than our implementation of the original method; we think that two of the reasons may be that their methods do not solve the exact problem as ours, and we did not adapt their whole method, but just a part of it (i.e. operators and rules). Furthermore, we have shown the importance of presenting all the results instead of the best one since it is probable that a method gets a feasible solution only once as seen in the results for the method CC (Table 4.11) where it only got one feasible solution out of 25 runs for the first instance.

5.1 Research Issues

In these kinds of problems, it is important to fully understand the constraints in order to evaluate the solutions, and know whether it shows any improvement or not. In our case, the first soft constraint: "each requirement should have at least μ double lessons a week" was not clear enough since there are two consecutive lessons in three or more consecutive lessons.

The way that solutions are represented is important too. Some authors represent the solution as a 1D array (Raghavjee & Pillay, 2013), whereas other authors represent their solution as a 2D array (Febrita & Mahmudy, 2017; Raghavjee & Pillay, 2013; Saptarini et al., 2018; Saviniec & Constantino, 2017; Skoullis et al., 2016; Sutar & Bichkar, 2017; Zhang et al., 2010). Also, these solutions might be represented as bits (Demirovic & Musliu, 2017). These different representations lead to different perturbation operators, and in order to apply these operators to the original ILS-TQ method, they had to be modified to work with how we represent the solutions.

Finally, it is possible that other methods solve this problem, but with different constraints, different values for each constraint, other resources (e.g. rooms), teachers are not previously assigned, etc. This means that their operators must take all this into consideration, and we had to take this into consideration in order to apply the modifications.

5.2 Future Work

The initial solution or group of solutions, that is going to be the start point of a metaheuristic, is important in order to get schedules with higher quality. However, it would be nice to propose modifications to the original initial solution algorithm used by the ILS-TQ. A better initial solution could improve the final solution given by any of our modifications.

Another important way to improve these techniques is how they accept the solutions in order to explore the space of solutions. Some techniques only accept better solutions, whereas others might accept worse solutions according to an acceptance criterion. Modifications to how the original ILS-TQ accepts its solutions might provide better results.

We wish to apply more modifications to the original perturbation operator, and see whether or not they show any improvement over the original method or our proposed methods. Furthermore, a combination of all these modifications (i.e. initial solutions, acceptance criteria, operators) need also to be explored. We applied our modifications separately, except for the hybrid SA and 2TQ.

Some of our methods got better % ARD over the original method in some instances. We also wish to know why it happens: we would like to know what patterns those instances follow compared to the rest, and what kind of solutions our methods generate. Similarly, we would also like to know why some instances only need one local search to find a feasible solution.

Parallelizing our methods using GPU would be nice, but it is not our top priority since a schedule is usually created once a year for a whole high school. There would be no problem even if the method takes one day to get a feasible schedule. However, an idea would be to parallelize the local search where each thread does a TQ move, and the best schedule is found by parallel reduction; once a schedule cannot be improved, the local search ends. This would change the original method from a First-Improvement to a Best-Improvement local search. This parallelized version is expected to get better results since it does more iterations and explores the solution space more compared to the original method because the parallel version chooses the best solution of the closest neighborhood whereas the original solution chooses the best first solution it finds (the sequential version can behave like the parallel version, but it will take a lot longer); furthermore, it would take less time to get a near-optimum solution. However, in order to verify this, more tests of the parallel implementation would be needed.

References

Bahrami, M., Bozorg-Haddad, O., & Chu, X. (2006). Advances optimization by nature-inspired algorithms. In (chap. Chapter 2: Cat Swarm Optimization (CSO) algorithm). Springer Nature Singapore.

Brito, S., Fonseca, G., & Haroldo, S. (2012). A simulated annealing based approach to the high school timetabling problem. *Electronic Notes in Discrete Mathematics*, 540-549.

Brito, S., Fonseca, G., Toffolo, T., Haroldo, S., & Marcone, S. (2012, December). A sa-vns approach for the high school timetabling problem. *Electronic Notes in Discrete Mathematics*, 169-176.

Carpaneto, G., & Toth, P. (1987). Primal-dual algorithms for the assignment problem. *Discrete Applied Mathematics*, *18*, 137-153.

Carr, J. (2014, May). *An introduction to genetic algorithms.*

Demirović, E., & Musliu, N. (2017). Maxsat-based large neighborhood search for high school timetabling. *Computers & Operations Research*, *78*, 172-180.

Demirovic, E., & Musliu, N. (2017). Modeling high school timetabling with bitvectors. *Annals of operations research*, 215-238.

Du, K., & Swamy, M. (2016). Simulated annealing. In B. Birkhauser (Ed.), *Techniques and algorithms inspired by nature - search and optimization by metaheuristics* (Vol. 1, p. 29-36).

Febrita, R., & Mahmudy, W. (2017). Modified genetic algorithm for high school timetable scheduling with fuzzy time window. *International Conference on Sustainable Information Engineering and Technology*, 88-92.

Fonseca, G., & Santos, H. (2013, June). Memetic algorithms to the high school timetabling problem. *Evolutionary computation*, 666-672.

Fonseca, G., & Santos, H. (2014, December). Variable neighborhood search based algorithms for high school timetabling. *Computers & Operations Research*, *52*, 203-208.

Fonseca, G., Santos, H., & Carrano, E. (2016, August). Late acceptance hill-climbing for high school timetabling. *Journal of Scheduling*, *19*, 453-465.

Fonseca, G., Santos, H., Toffolo, T., Brito, S., & Souza, M. (2012, August). A sa-ils approach for the high school timetabling problem. In D. Kjenstad, A. Riise, T. Nordlander, B. McCollum, & E. Burke (Eds.), *Practice and theory of automated timetabling: Proceedings of the 9th international conference on the practice and theory of automated timetabling, patat 2012* (p. 493-496).

Fonseca, G., Santos, H., Toffolo, T., Brito, S., & Souza, M. (2016, April). Goal solver: a hybrid local search based solver for high school timetabling. *Annals of Operation Research*, *239*, 77-97.

Gendreau, M., & Potvin, J. (2005, November). Metaheuristics in combinatorial optimization. *Annals of Operations Research*, *140*, 189-213.

Glover, F. (1989). Tabu search, part i. *ORSA Journal on Computing*, 190-206.

Herbawi, W. (2014, November). *What is the major difference between heuristic, hyper huristic and meta heuristics?, and their relationship with global optimization methods?* Available: `https://www.researchgate.net/post/What_is_the_major_difference_between_heuristic_hyper_heuristic_and_meta_heuristics_and_their_relationship_with_global_optimization_methods`. ([Accessed: April 10, 2018])

Katsaragakis, I., Tassopoulos, I., & Beligiannis, G. (2015). A comparative study of modern heuristics on the school timetabling problem. *Algorithms*, *8*, 723-742.

Kingston, J. (2015, December). *The khe high school timetabling engine.*

Minh, K., Thanh, N., Trang, K., & Hue, N. (2010). Using tabu search for solving a high school timetabling problem. *Intelligent Information and Database Systems*, 305-313.

Odeniyi, O., Omidiora, E., Olabiyisi, S., & Aluko, J. (2015, January). Development of a modified simulated annealing to school timetabling problem. *International Journal of Applied Information Systems*, *8*.

Post, G., Gaspero, L. D., Kingston, J., McCollum, B., & Schaerf, A. (2016). The third international timetabling competition. *Annals of Operations Research*, *239*, 69-75.

Raghavjee, R., & Pillay, N. (2013). A comparative study of genetic algorithms using a direct and indirect representation in solving the african school timetbling problem. In *Proceedings of the 2013 orssa annual conference* (p. 31-39).

Saptarini, N., Suasnawa, I., & Ciptayani, P. (2018). Senior high school course scheduling using genetic algorithm. *The 2nd International Joint Conference on Science and Technology*.

Saviniec, L., Aparecido, A., & Romao, W. (2013). Solving the high school timetabling problem to optimality by using ils algorithms. *Simpósio Brasileiro de Pesquisa Operacional*.

Saviniec, L., & Constantino, A. (2017, November). Effective local search algorithms for high school timetabling problems. *Applied Soft Computing*, *60*, 363-373.

Skoullis, V., Tassopoulos, I., & Beligiannis, G. (2016). Solving the high school timetabling problem using a hybrid cat swarm optimization based algorithm. *Applied Soft Computing*.

Sutar, S., & Bichkar, R. (2017, June). High school timetabling using tabu search and partial feasibility preserving genetic algorithm. *International Journal of Advances in Engineering & Technology*, *10*, 421-428.

Templatetypedef. (2014, December). *I need to solve an np-hard problem. is there hope?* Available: `https://stackoverflow.com/questions/27455585/i-need-to-solve-an-np-hard-probl%D0%B5m-is-there-hope`. ([Accessed: April 05, 2018])

Yousef, M., Khader, A., Kheiri, A., & Ozcan, E. (2013, November). A two stage approach for high school timetabling. *International Conference on Neural Information Processing*, 66-73.

Zhang, D., Liu, Y., M'Hallah, R., & Leung, S. (2010). A simulated annealing with a new neighborhood structure based algorithm for high school timetabling problems. *European Journal of Operational Research*, 550-558.